

APMA 922: Homework Set 04

Joseph Lucero

November 11, 2019

Problem 1

Part A

In order to apply TChebyshev spectral methods to solve BVPs we must scale and translate the domain variable $x \in [0, 1]$ of the given BVP into the variable $z \in [-1, 1]$. To do this we let

$$z = 2x - 1,$$

and thus have that

$$x = \frac{z + 1}{2}.$$

So the variable $z \in [-1, 1]$ and thus we can apply TChebyshev spectral methods on this variable. With this transformation the differentials are also transformed. In particular we have

$$dx = \frac{1}{2} dz,$$

which then transforms the definition of the derivative

$$\frac{d}{dx} \rightarrow 2 \frac{d}{dz},$$

and similarly for the second derivative

$$\frac{d^2}{dx^2} \rightarrow 4 \frac{d^2}{dz^2}.$$

Thus the differential equation in variable x is recast into the differential equation in variable z ,

$$\left[4\varepsilon \frac{d^2}{dz^2} + 2 \frac{d}{dz} \right] y(z) = -\exp\left(\frac{1-z}{2}\right), \quad -1 < z < 1, \quad y(-1) = y(1) = 0. \quad (1)$$

Figure 1 shows, in the left subplot, the solution acquired from the spectral method in the black curve. As can be seen the computed solution agrees very well with the exact solution that is shown with the red curve. The right subplot of Fig. 1 we observe that for $\varepsilon = 1$, for $N = 16$, the error of the solution is already as small as it can be. We also note that the N , for which the error is minimum, is increased the smaller ε is. This, as we saw before, is because a boundary layer appears as $\varepsilon \rightarrow 0$ and thus requires more points to resolve. We observe that it requires about an accuracy of 10^{-5} is achieved for $N = 5$.

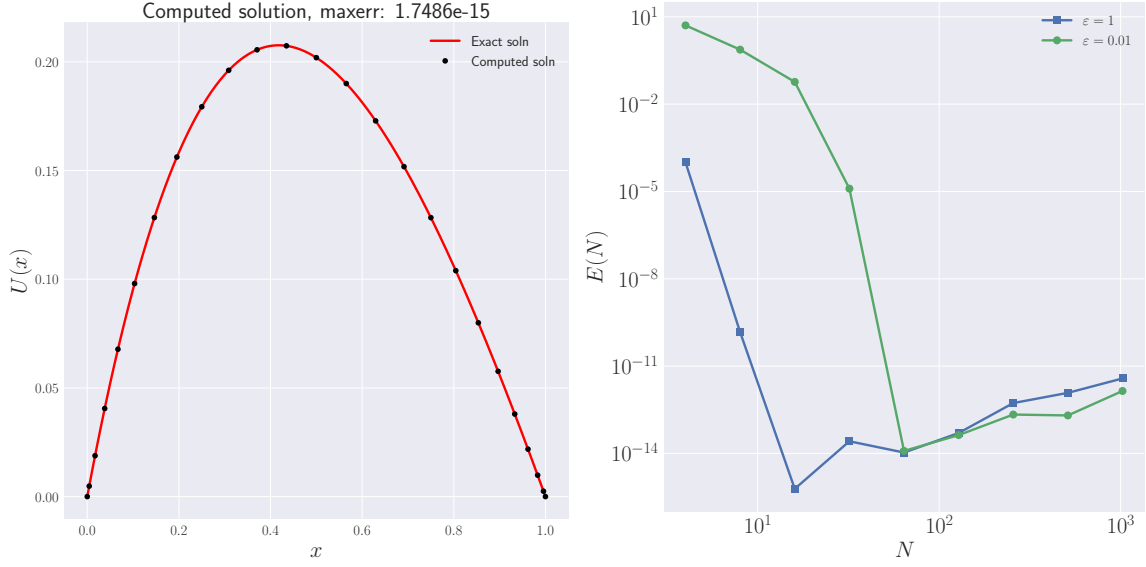


Figure 1: [Left] Exact (red line) and computed (black points) solution as a function of x for $N = 24$ and $\varepsilon = 1$. [Right] Infinity-norm error of the computed solution for $\varepsilon = 1$ (blue line) and $\varepsilon = 0.01$ (green line) as a function of the number of intervals N .

Part B

As above, we rescale the domain variable $x \in [-2, 1]$ to a variable $z \in [-1, 1]$. To do this we let

$$z = \frac{2x + 1}{3},$$

and thus have that

$$x = \frac{3z - 1}{2}.$$

So the variable

$$z \in [-1, 1]$$

and we can apply TChebyshev spectral methods on this variable. With this transformation the differentials are also transformed. In particular we have

$$dx = \frac{3}{2} dz,$$

which then transforms the definition of the derivative

$$\frac{d}{dx} \rightarrow \frac{2}{3} \frac{d}{dz},$$

and similarly for the second derivative

$$\frac{d^2}{dx^2} \rightarrow \frac{4}{9} \frac{d^2}{dz^2}.$$

Thus the differential equation in variable x is recast into the differential equation in variable z ,

$$\left[\frac{4}{9} \frac{d^2 u}{dz^2} - 4(3z - 1) \right] u(z) = -(3z - 1)^2, \quad -1 < z < 1, \quad u(-1) = 0, \quad u'(1) = 2. \quad (2)$$

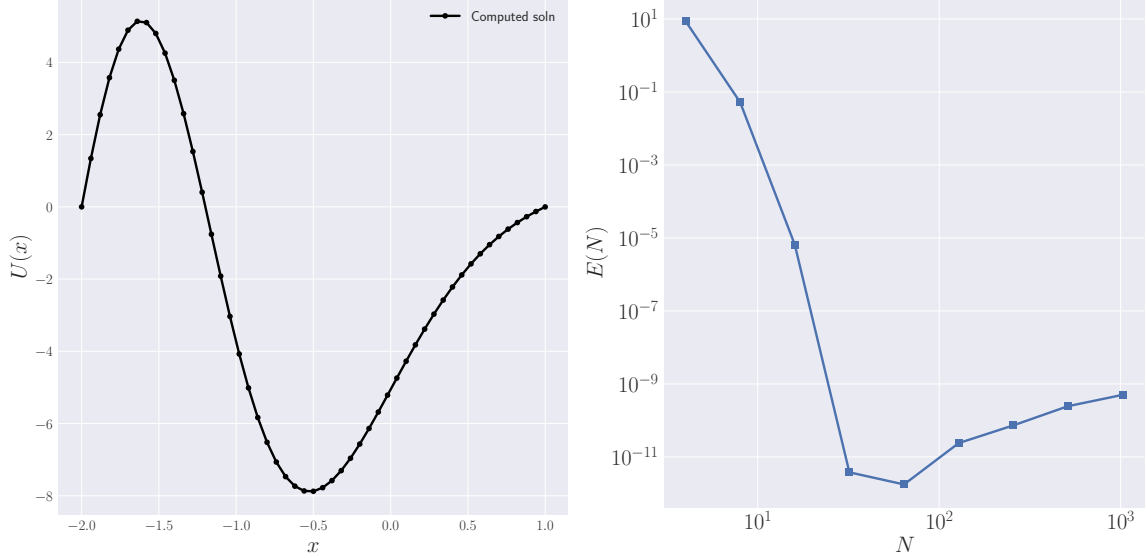


Figure 2: [Left] Computed solution as a function of x . [Right] Infinity-norm error of the computed solution as a function of the number of intervals N .

Figure 2 shows, in the left subplot, the solution acquired from the spectral method in the black curve. The right subplot of Fig. 2 we observe that, as we expect, we attain spectral accuracy for this method. To get an accuracy of 10^{-4} we observe that we need $N = 10$ number of intervals.

Problem 2

Part A

Using the definition of the differentiation matrix given that we derived in the last assignment,

$$D_{ij} = \begin{cases} \frac{a_i}{a_j(x_i - x_j)}, & i \neq j \\ \sum_{k=0, k \neq j}^N (x_j - x_k), & i = j. \end{cases} \quad (3)$$

we implemented this differentiation matrix in `A4Q2.ipynb`. Using the TChebyshev extreme points $\{x_j\}$ in the definition above returns the TChebyshev differentiation matrix as we expect. Demonstration of this test in one of the cells in the notebook.

Part B

This is implemented in `A4Q2.ipynb`. Since the algorithm in `gauss.m` yields the N zeros of the Legendre polynomial, to generate our grid in the interval $[-1,1]$, the interior points are generated from $N - 1$ zeros of the Legendre polynomial. The endpoints constitute the remaining two points that bring the total number of points in the grid to $N + 1$, consistent with the output of the `cheb.m` algorithm.

Part C

Figure 3 is a recreation of Output 11 from [Tr]. Here the function $u(x)$ is evaluated on the TChebyshev grid and the TChebyshev differentiation matrix is used to compute the spectral derivative. Figure 4 is the recreation of Output 11 except using Legendre points to evaluate the function $u(x)$ and the Legendre differentiation matrix to compute the spectral derivative.

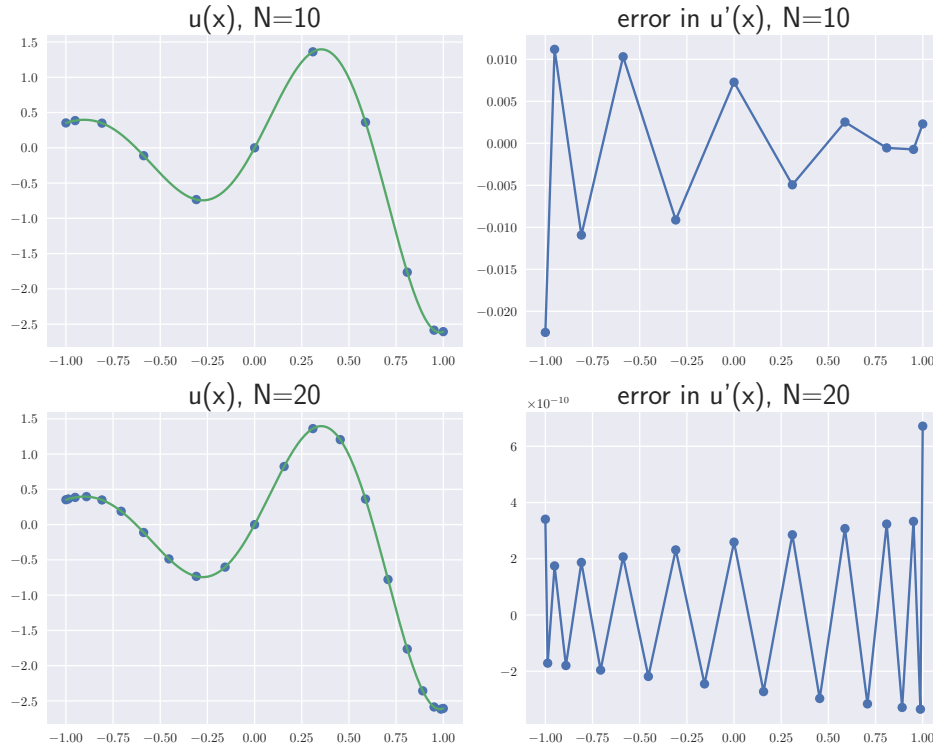


Figure 3: [Left column] Function $u(x)$ evaluated on the TChebyshev grid for $N = 10$ (top left subplot) and $N = 20$ (bottom left subplot) as a function of x . Associated infinity-norm error of the spectral derivative $u'(x)$ for $N = 10$ (top right subplot) and $N = 20$ (bottom right subplot) as a function of x .

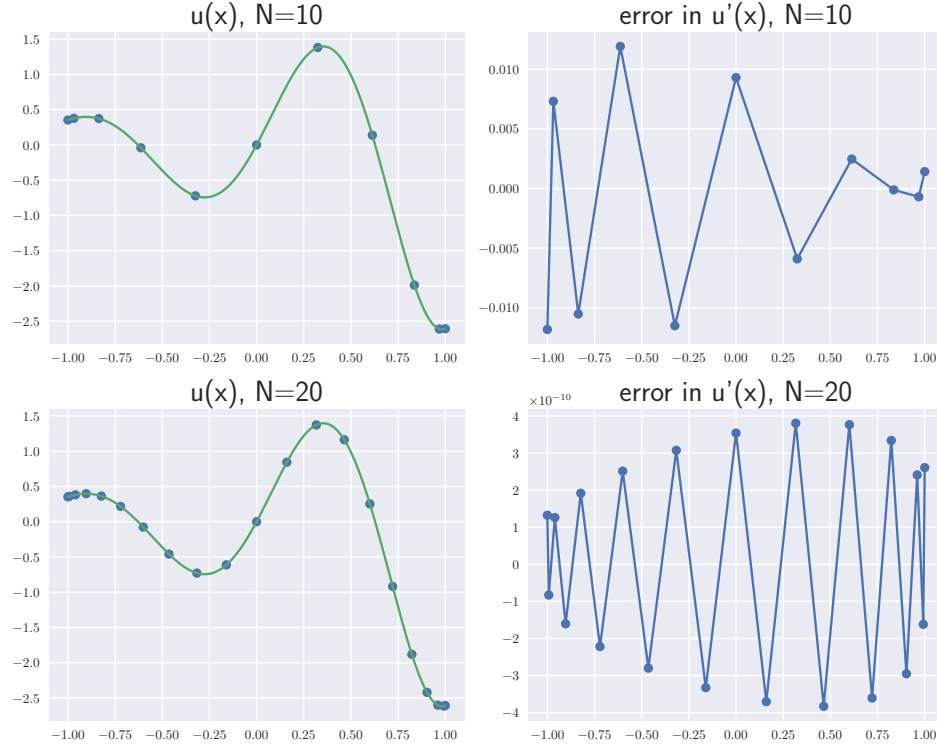


Figure 4: [Left column] Function $u(x)$ evaluated on the Legendre grid for $N = 10$ (top left subplot) and $N = 20$ (bottom left subplot) as a function of x . Associated infinity-norm error of the spectral derivative $u'(x)$ for $N = 10$ (top right subplot) and $N = 20$ (bottom right subplot) as a function of x .

Part D

Figure 5 shows the computed solution for the two BVPs that we are asked to solve. In the left subplot is the computed solution for the differential equation $u''(x) = e^{4x}$ acquired from utilizing the TChebyshev (red solid line) and the Legendre (black dashed line) spectral differentiation matrix. Since for this BVP $x \in [-1, 1]$, no rescaling of the domain was necessary.

In the right subplot the computed solution for the differential equation $u''(x) - 8xu = -4x^2$ acquired from utilizing the TChebyshev (red solid line) and the Legendre (black dashed line) spectral differentiation matrix. The scaling of the domain used in Question 1 Part B was used once again to solve the BVP on the domain $z \in [-1, 1]$.

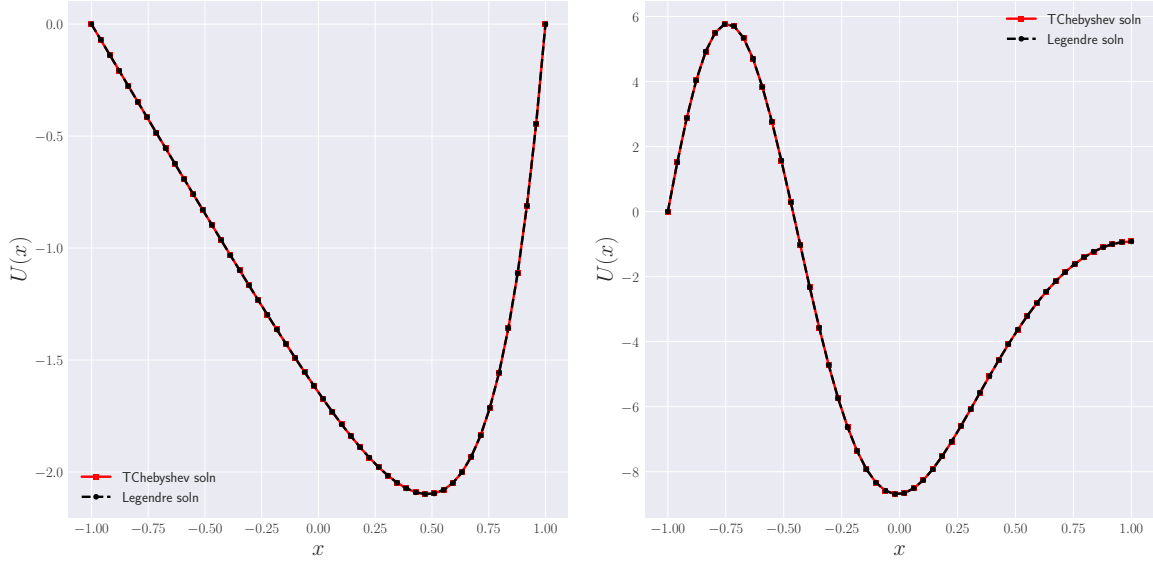


Figure 5: [Left] Computed solution, $U(x)$, of the differential equation $u''(x) = e^{4x}$ as a function of x . [Right] Computed solution, $U(x)$, of the differential equation $u''(x) - 8xu = -4x^2$ as a function of x . Red solid lines are solution computed using Tchebyshev differentiation matrices. Black dashed lines are solution computed using Legendre differentiation matrices.

Problem 3

Part A

As in the discussion we have that the forward Euler method for this problem, with constant step size k produces iterates of the form $U^n = (1 + k\lambda)^n$. As in the problem discussion we have use $N = 200$ with steps of size $k = 0.01$. This yields the computed solution

$$U^N = (1 + (0.01)10)^{200} \quad (4a)$$

$$= (1.1)^{200} \quad (4b)$$

$$\approx 1.90 \times 10^8. \quad (4c)$$

The true solution is $u(T) = e^{20} \approx 4.85 \times 10^8$. Thus we observe that for this case the error bound becomes a more reasonable estimate of the error. The Forward-Euler time stepping scheme is implemented in `A4Q3.ipynb`.

Part B

The general explicit one-step method takes the form

$$U^{n+1} = U^n + k\Psi(U^n, t_n, k).. \quad (5)$$

In this case, since Ψ is linear in u , we have that

$$\Psi(U^n, t_n, k) - \Psi(U^n, t_n, k) = \lambda [U^n - u(t_n)] = \lambda E^n. \quad (6)$$

Given this we can then write the global error as

$$E^{n+1} = E^n + k [\Psi(U^n, t_n, k) - \Psi(U^n, t_n, k)] - k\tau^n \quad (7a)$$

$$E^{n+1} = (1 + k\lambda)E^n - k\tau^n. \quad (7b)$$

By induction we have that

$$E^n = (1 + k\lambda)^n E^0 - k \sum_{m=1}^n (1 + k\lambda)^{n-m} \tau^{m-1} \quad (8)$$

Taking absolute values, applying triangle inequality, we have

$$|E^n| \leq |1 + k\lambda|^n |E^0| + k \sum_{m=1}^n |1 + k\lambda|^{n-m} |\tau^{m-1}| \quad (9)$$

We thus identify the Lipschitz constant of Ψ as $M = |1 + k\lambda|$. Using that $M \leq 1$,

$$|E^n| \leq |E^0| + k \sum_{m=1}^n |\tau^{m-1}| \quad (10a)$$

$$\leq |E^0| + t \|\tau\|_\infty, \quad (10b)$$

where we have that $N = T/k$ is the number of time steps needed to reach time T and where we upper bounded the sum of the truncation errors with the maximum truncation error

$$\|\tau\|_\infty = \max_{0 \leq n \leq N-1} |\tau^n|,$$

as well as using that $t = nk$ from the definition of the total number of time steps. Now for $t \leq T$ we can write

$$|E^n| \leq |E^0| + T \|\tau\|_\infty, \quad (11)$$

and stipulating the $|E^0| = 0$ then we have

$$|E^n| \leq T \|\tau\|_\infty, \quad (12)$$

as desired.

Part C

We show above that we can take $M = |1 + k\lambda|$, and so if we now apply it to the given case we obtain the bound

$$|E^n| \leq 2 \cdot \frac{1}{2} k \|u''\|_\infty = 100 \quad (13)$$

which indeed is much better than the one obtained previously as it lacks the exponential term.

Part D

i)

Following the hint,

$$|u - u^*| = |\Phi(v) - \Phi(v^*)| \quad (14a)$$

$$= |g^{-1}(v) - g^{-1}(v^*)| \quad (14b)$$

$$= |g^{-1}([u - kf(u)]) - g^{-1}([u^* - kf(u^*)])| \quad (14c)$$

$$= |v - kf(v) - v^* + kf(v^*)| \quad (14d)$$

$$\leq |v - v^*| + k|f(v) - f(v^*)| \quad (14e)$$

$$\leq (1 + kL)|v - v^*|, \quad (14f)$$

where in going from (14d) to (14e) we used the triangle inequality and going from (14e) to (14f) we used the Lipschitz continuity of f . If $0 < k < 1/L$ then this can be further upper bounded by

$$|u - u^*| = |\Phi(v) - \Phi(v^*)| \leq \frac{1}{1 - kL}|v - v^*| \quad (15)$$

Thus we see that Φ is Lipschitz continuous with Lipschitz constant $(1 - kL)^{-1}$.

ii)

The Backward-Euler discretization of a initial value problem $u' = f(u, t)$ is given by

$$\frac{U^{n+1} - U_n}{k} = f(U^{n+1}), \quad (16)$$

and thus can be recast into the form

$$U^n = U^{n+1} - kf(U^{n+1}). \quad (17)$$

Then, if we define as in the discussion of the previous part $g(U^{n+1}) := U^{n+1} - kf(U^{n+1})$ so we have

$$U^n = g(U^{n+1}) \quad (18)$$

and subsequently, defining $\Phi(U^n) := g^{-1}(U^n)$, we see that if we apply this function Φ on both sides of the equation we have that

$$\Phi(U^n) = \Phi(g(U^{n+1})) \quad (19a)$$

$$= U^{n+1}. \quad (19b)$$

Then we see the backward Euler scheme takes the form desired. Since from i) we have that Φ is Lipschitz continuous with Lipschitz constant $\mathcal{L} \leq 1$ and we have from Part B that equations of the form (19) can have the error be bounded uniformly of the step size thus the backward Euler converges.

Problem 4

Part A

One way to derive the Adams family of methods is by writing

$$u(t_{n+r}) = u(t_{n+r-1}) + \int_{t_{n+r-1}}^{t_{n+r}} ds f(u(s)) \quad (20)$$

and using a quadrature rule to approximate

$$\int_{t_{n+r-1}}^{t_{n+r}} dt f(u(t)) \approx k \sum_{j=1}^{r-1} \beta_j f(u(t_{n+j})). \quad (21)$$

This quadrature rule can be derived by interpolating $f(u(t))$ by a polynomial $p(t)$ of degree $r-1$ at the points $t_n, t_{n+1}, \dots, t_{n+r-1}$ and then integrating the interpolating polynomial. In this case we are interested in interpolating $f(u(t))$ through the points $t-k, t, t+k$.

The interpolating polynomial in the Lagrange basis is given by,

$$p_2(t) = L(t) = \sum_{j=0}^k f^j l_j(t). \quad (22)$$

with basis functions,

$$l_j(t) = \prod_{\substack{0 \leq m \leq k \\ m \neq j}} \frac{t - t_m}{t_j - t_m}. \quad (23)$$

The interpolant is then

$$p_2(t) = f^{n-1} \left(\frac{t - t_n}{t_{n-1} - t_n} \right) \left(\frac{t - t_{n+1}}{t_{n-1} - t_{n+1}} \right) + f^n \left(\frac{t - t_{n-1}}{t_n - t_{n-1}} \right) \left(\frac{t - t_{n+1}}{t_n - t_{n+1}} \right) \\ + f^{n+1} \left(\frac{t - t_{n-1}}{t_{n+1} - t_{n-1}} \right) \left(\frac{t - t_n}{t_{n+1} - t_n} \right) \quad (24)$$

Using that $t_{n-1} = t_n - k, t_{n+1} = t_n + k$, the interpolating polynomial is then we have

$$p_2(t) = \frac{(-2f_n + f_{n-1} + f_{n+1})}{2k^2} t^2 + \frac{(4f^n t_n - f^{n-1}k - 2f^{n-1}t_n + f^{n+1}k - 2f^{n+1}t_n)}{2k^2} t \\ + \frac{2f^n k^2 - 2f^n t_n^2 + f^{n-1}k t_n + f^{n-1}t_n^2 - f^{n+1}k t_n + f^{n+1}t_n^2}{2k^2}. \quad (25)$$

3-step Adams-Bashforth method

Integrating from time step $t_n + k$ to $t_n + 2k$ we then have

$$\int_{t_{n+1}}^{t_{n+2}} dt p_2(t) = \int_{t_n+k}^{t_n+2k} dt p_2(t) \quad (26a)$$

$$\begin{aligned} &= \frac{(-4f^n + 2f^{n-1} + 2f^{n+1})}{12k^2} t^3 + \frac{(12f^n t_n - 3f^{n-1}k - 6f^{n-1}t_n + 3f^{n+1}k - 6f^{n+1}t_n)}{12k^2} t^2 \\ &\quad + \frac{(12f^n k^2 - 12f^n t_n^2 + 6f^{n-1}kt_n + 6f^{n-1}t_n^2 - 6f^{n+1}kt_n + 6f^{n+1}t_n^2)}{12k^2} t \Big|_{t_n+k}^{t_n+2k} \end{aligned} \quad (26b)$$

$$= \frac{k}{12} (5f^{n-1} - 16f^n + 23f^{n+1}). \quad (26c)$$

Thus we obtain the three-step Adams-Bashforth scheme

$$U^{n+2} = U^{n+1} + \int_{t_{n+1}}^{t_{n+2}} dt p_2(t) \quad (27a)$$

$$= U^{n+2} + \frac{k}{12} (5f^{n-1} - 16f^n + 23f^{n+1}), \quad (27b)$$

which we can get into the form given in the question by simply shifting the index by 1,

$$U^{n+3} = U^{n+2} + \frac{k}{12} (5f^n - 16f^{n+1} + 23f^{n+2}). \quad (28)$$

2-step Adams-Moulton method

Integrating the interpolating polynomial from t_n to $t_n + k$ instead we now have

$$\int_{t_n}^{t_{n+1}} dt p_2(t) = \int_{t_n}^{t_n+k} dt p_2(t) \quad (29a)$$

$$\begin{aligned} &= \frac{(-4f^n + 2f^{n-1} + 2f^{n+1})}{12k^2} t^3 + \frac{(12f^n t_n - 3f^{n-1}k - 6f^{n-1}t_n + 3f^{n+1}k - 6f^{n+1}t_n)}{12k^2} t^2 \\ &\quad + \frac{(12f^n k^2 - 12f^n t_n^2 + 6f^{n-1}kt_n + 6f^{n-1}t_n^2 - 6f^{n+1}kt_n + 6f^{n+1}t_n^2)}{12k^2} t \Big|_{t_n}^{t_n+k} \end{aligned} \quad (29b)$$

$$= \frac{k}{12} (-f^{n-1} + 8f^n + 5f^{n+1}). \quad (29c)$$

And so we obtain the two-step Adams-Moulton scheme

$$U^{n+1} = U^n + \int_{t_n}^{t_{n+1}} dt p_2(t) \quad (30a)$$

$$= U^n + \frac{k}{12} (-f^{n-1} + 8f^n + 5f^{n+1}), \quad (30b)$$

which can be translated into the form given by a simple shift of index to the desired form

$$U^{n+2} = U^{n+1} + \frac{k}{12} (-f^n + 8f^{n+1} + 5f^{n+2}). \quad (31)$$

2-step backward differentiation formula (BDF) method

To acquire the backward differentiation formula, rather than fit a polynomial through the right-hand side $f(u(t), t)$, we instead fit a polynomial interpolant through u at points $t_n - k, t_n, t_n + k$ and search for the linear combination that gives the approximation to the derivative at $t_n + k$. To make this a bit easier, we instead fit the polynomial through points $t_n - 2k, t_n - k, t_n$ and search for the approximation to the derivative at t_n . So we are interested in finding the coefficients of the linear combination that gives

$$u'(t_n) = au(t_n) + bu(t_n - k) + cu(t_n - 2k). \quad (32)$$

Taylor expanding the right hand side in powers of k we have

$$\begin{aligned} au(t_n) + bu(t_n - k) + cu(t_n - 2k) &= (a + b + c)u(t_n) - (b + 2c)ku'(t_n) + \left(\frac{1}{2}b + 2c\right)k^2u''(t_n) \\ &\quad - \left(\frac{1}{6}b + \frac{4}{3}c\right)k^3u'''(t_n) + \mathcal{O}(k^4). \end{aligned} \quad (33)$$

Therefore we have a system of equations

$$a + b + c = 0 \quad (34a)$$

$$b + 2c = -\frac{1}{k} \quad (34b)$$

$$\frac{1}{2}b + 2c = 0, \quad (34c)$$

which has the solution

$$a = \frac{3}{2k}, \quad b = -\frac{4}{2k}, \quad c = \frac{1}{2k}.$$

Thus

$$u'(t_n) = \frac{1}{2k} [3u(t_n) - 4u(t_n - k) + u(t_n - 2k)], \quad (35)$$

and using that $u'(t_n) = f(u(t_n), t_n)$ then we reacquire the BDF formula

$$3u(t_n) - 4u(t_n - k) + u(t_n - 2k) = 2kf(u(t_n), t_n), \quad (36)$$

or equivalently

$$3U^n - 4U^{n-1} + U^{n-2} = 2kf^n. \quad (37)$$

To get it into the form given then we shift the index by 2,

$$3U^{n+2} - 4U^{n+1} + U^n = 2kf^{n+2}. \quad (38)$$

Part B

AB3 formula

The general LTE formula for LMMs is given by

$$\begin{aligned}
\tau(t_{n+r}) = & \frac{1}{k} \left(\sum_{j=0}^r \alpha_j \right) u(t_n) + \left(\sum_{j=0}^r (j\alpha_j - \beta_j) \right) u'(t_n) \\
& + k \left(\sum_{j=0}^r \left(\frac{1}{2} j^2 \alpha_j - j\beta_j \right) \right) u''(t_n) \\
& + k^2 \left(\sum_{j=0}^r \left(\frac{1}{6} j^3 \alpha_j - \frac{1}{2} j^2 \beta_j \right) \right) u'''(t_n) \\
& + k^3 \left(\sum_{j=0}^r \left(\frac{1}{24} j^4 \alpha_j - \frac{1}{6} j^3 \beta_j \right) \right) u''''(t_n) \\
& + \dots + k^{q-1} \left(\sum_{j=0}^r \left(\frac{1}{q!} j^q \alpha_j - \frac{1}{(q-1)!} j^{q-1} \beta_j \right) \right) u^{(q)}(t_n) + \dots
\end{aligned} \tag{39}$$

To derive the AB3 formula we impose the regular consistency conditions,

$$\sum_{j=0}^3 \alpha_j = 0 \quad \text{and} \quad \sum_{j=0}^3 (j\alpha_j - \beta_j) = 0, \tag{40}$$

as well as that

$$\sum_{j=0}^3 \left(\frac{1}{2} j^2 \alpha_j - j\beta_j \right) = 0 \quad \text{and} \quad \sum_{j=0}^3 \left(\frac{1}{3} j^3 \alpha_j - \frac{1}{2} j^2 \beta_j \right) = 0 \tag{41}$$

and given that $\alpha_3 = 1$, $\alpha_2 = -1$, $\alpha_1 = \alpha_0 = \beta_3 = 0$, solving these three equations for the β_r yields, as expected

$$\beta_2 = \frac{23}{12}, \beta_1 = -\frac{16}{12}, \beta_0 = \frac{5}{12}. \tag{42a}$$

Here $\beta_3 = 0$ since the method is explicit. Substituting these set of α_r and β_r values into the equations shows that these do indeed satisfy the equations.

AM2 formula

To derive the AM2 formula we impose the conditions above,

$$\sum_{j=0}^2 \alpha_j = 0, \quad \sum_{j=0}^2 (j\alpha_j - \beta_j) = 0 \quad \text{and} \quad \sum_{j=0}^2 \left(\frac{1}{2} j^2 \alpha_j - j\beta_j \right) = 0, \tag{43}$$

as well as that

$$\sum_{j=0}^2 \left(\frac{1}{6} j^3 \alpha_j - \frac{1}{2} j^2 \beta_j \right) = 0, \quad (44)$$

and given that $\alpha_2 = 1$, $\alpha_1 = -1$, $\alpha_0 = 0$, solving these three equations for set of the β_r yields, as expected

$$\beta_2 = \frac{5}{12}, \beta_1 = \frac{8}{12}, \beta_0 = -\frac{1}{12}. \quad (45)$$

Like above, substituting these set of α_r and β_r values into the equations shows that these do indeed satisfy the equations.

BDF2 formula

If instead we impose the conditions as in the AM2 formula but instead impose that $\beta_2 = 1$, $\beta_1 = \beta_0 = 0$ and solve for the α_r we instead acquire that

$$\alpha_2 = \frac{3}{2k}, \alpha_1 = -\frac{4}{2k}, \alpha_0 = \frac{1}{2k} \quad (46)$$

and thus reacquire the BDF formula.

Part C

The characteristic polynomials for the Adams-Bashforth 3-step method are given by

$$\rho(\zeta) = \sum_{j=0}^3 \alpha_r \zeta^r = \zeta^3 - \zeta^2 \quad (47)$$

$$\sigma(\zeta) = \sum_{j=0}^2 \beta_r \zeta^r = \frac{23}{12} \zeta^2 - \frac{16}{12} \zeta + \frac{5}{12}, \quad (48)$$

and the stability polynomial is then

$$\pi(\zeta; z) = \rho(\zeta) - z\sigma(\zeta) = \zeta^3 - \zeta^2 - z \left(\frac{23\zeta^2}{12} - \frac{4\zeta}{3} + \frac{5}{12} \right). \quad (49)$$

As we saw above the leading order for this method is $\mathcal{O}(k^3)$ as we eliminated the first four terms in the general truncation error (39). The leading term in the truncation error is then

$$k^3 \sum_{j=0}^3 \left(\frac{1}{24} j^4 \alpha_j - \frac{1}{6} j^3 \beta_j \right) u''''(t_n) = \frac{3}{8} k^3 u''''(t_n). \quad (50)$$

To check zero stability, we check the roots of the characteristic polynomial

$$\rho(\zeta) = \zeta^2(\zeta - 1), \quad (51)$$

and so we see that the zeros of this polynomial are $\zeta = 0$ and $\zeta = 1$, and thus we see that it satisfies the root condition and so is zero-stable.

Part D

Porting [LV] code `makeplotBL.m` into Python, we acquire the boundary locus plot for AB3 shown in Fig. ??.

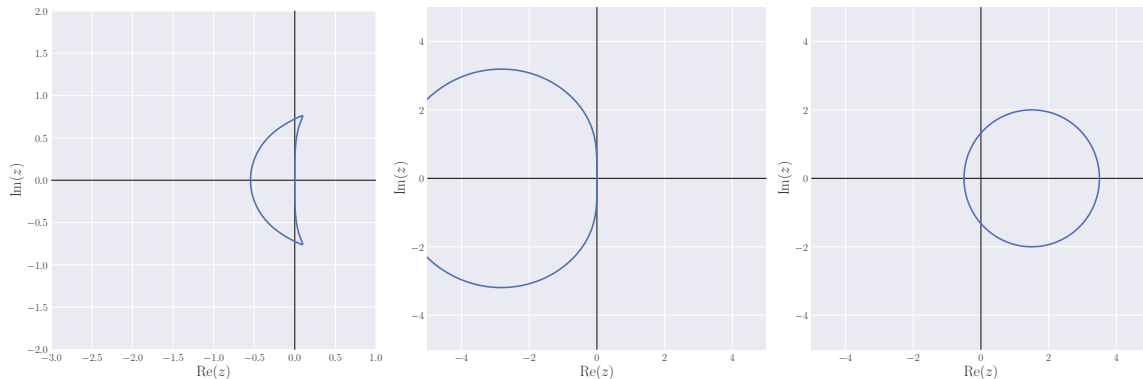


Figure 6: Plotting $z = \rho/\sigma$ on the complex plane. [Left subplot] The region *inside* the region outlined is the stability-region for the AB3 method. [Middle subplot] The region *inside* the outlined is the stability-region for the AM2 method. [Right subplot] The region *exterior* to the region outlined is the stability-region for the BDF2 method.

Part E

We implement the AB3 method in `A4Q4.ipynb` and use an RK4 method to initialize the first 3 points. We compute the solution for the equation $u' = \lambda u$ with $\lambda = -1$ for a variety of k .

The left subplot of Fig. 7, we observe behaviours in the solution consistent with the boundary loci computed in Fig. 6. In particular we see that for $k = 0.5$, as we approach the boundary of the stability region, the solution begins to deviate from the solution. For $k = 0.75$ when we have passed the apex of the stability region, we see that the computed solution, really deviates from the exact solution in an exponential function. The lines extend out of the plot as the plot is a log-lin plot and whenever a value becomes negative, it is not well captured in this type of plot.

The right subplot of Figure 7 shows the error as a function of step size k when k is inside the stability region. We thus confirm that this method is indeed third-order within the stability region.

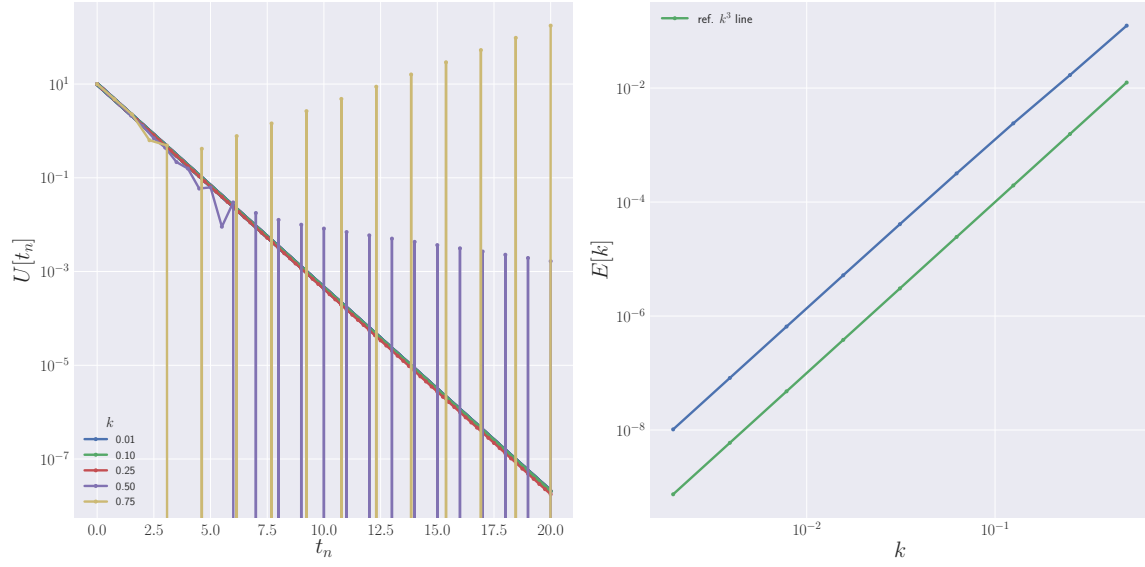


Figure 7: [Left subplot] Computed solution, $U[t_n]$, to the equation $u' = \lambda u$ using the AB3 method, as a function of time step t_n . [Right subplot] Infinity-norm error between the solution computed using AB3 and the true solution as a function of time step size k .

Problem 5

Part A

AB2 method

The interpolating polynomial is given by

$$p(t) = \frac{t - t_{n-1}}{t_n - t_{n-1}} f^n + \frac{t - t_n}{t_{n-1} - t_n} f^{n-1}. \quad (52)$$

Integrating this polynomial from time t_n to time t_{n+1} we have

$$\int_{t_n}^{t_{n+1}} dt p(t) = \frac{1}{k} \left\{ \left(\frac{f_n}{2} - \frac{f_{n-1}}{2} \right) t^2 + [f^n(k - t_n) + f^{n-1}t_n] t \right\} \Big|_{t_n}^{t_{n+k}} \quad (53a)$$

$$= \frac{k}{2} (3f^n - f^{n-1}). \quad (53b)$$

Therefore, the 2-step Adams-Bashforth method is given by

$$U^{n+1} = U^n + \frac{k}{2} (3f(U^n, t_n) - f(U^{n-1}, t_{n-1})). \quad (54)$$

To find its local truncation error we first define the shorthand

$$f^{(n,m)} = \frac{\partial^2}{\partial u^n \partial t^m} f(u(t), t).$$

We then substitute in the true solution u into the above equation of the method (54) and Taylor expand all the things,

$$\tau(t_n) = \frac{u(t_n + k) - u(t_n)}{k} - \frac{1}{2} (3f(u(t_n), t_n) - f(u(t_n - k), t_n - k)) \quad (55a)$$

$$= \frac{k^2}{12} \left(3u''(t_n)f^{(1,0)}(u(t_n), t_n) + 3u'(t_n)^2 f^{(2,0)}(u(t_n), t_n) \right. \\ \left. + 6u'(t_n)f^{(1,1)}(u(t_n), t_n) + 3f^{(0,2)}(u(t_n), t_n) + 2u^{(3)}(t_n) \right) + \mathcal{O}(k^3) \quad (55b)$$

$$= \frac{5}{12} k^2 u^{(3)}(t_n) + \mathcal{O}(k^3), \quad (55c)$$

where in going from the penultimate step to the last we used that

$$u^{(3)}(t) = \frac{d}{dt} \left\{ \frac{d}{dt} [u'(t)] \right\} \quad (56a)$$

$$= \frac{d}{dt} \left\{ \frac{d}{dt} [f(u(t), t)] \right\} \quad (56b)$$

$$= \frac{d}{dt} [f(u(t), t)f^{(1,0)}(u(t), t) + f^{(0,1)}(u(t), t)] \quad (56c)$$

$$= u''(t)f^{(1,0)}(u(t), t) + u'(t)^2 f^{(2,0)}(u(t), t) + 2u'(t)f^{(1,1)}(u(t), t) + f^{(0,2)}(u(t), t), \quad (56d)$$

to simplify the expression.

Explicit RK2 method

To derive the 2nd-order explicit two-stage Runge-Kutta method with $c_1 = 0$, $c_2 = 1$, we approximate $f(u(t_n + s), t_n + s)$ by its (linear) interpolant through the points f^n and f^* . This polynomial is given by

$$p(t) = \frac{t - t^*}{-k} f^n + \frac{t - t_n}{k} f^*, \quad (57)$$

and integrating from t_n to t_{n+1} gives,

$$\int_{t_n}^{t_{n+1}} dt p(t) = \frac{1}{k} \left\{ \left(\frac{f^*}{2} - \frac{f_n}{2} \right) t^2 + [f^n(k + t_n) - f^* t_n] t \right\} \Big|_{t_n}^{t_n+k} \quad (58a)$$

$$= \frac{k}{2} [f^n + f^*] \quad (58b)$$

$$= \frac{k}{2} [f(U^n, t_n) + f(U^*, t_n + k)]. \quad (58c)$$

Therefore, the explicit RK2 method is given by

$$U^{n+1} = U^n + \frac{k}{2} [f(U^n, t_n) + f(U^*, t_n + k)], \quad (59)$$

which is just Heun's method as we have already seen.

Part B

Considering the ODE,

$$u' = g(u, t) = \lambda u + f(u, t) \quad (60)$$

we apply Forward Euler (FE), AB2, and RK2 to it.

AB1 = FE

Applying the FE discretization (also equivalent to a 1-stage Adams-Bashforth method) we have that,

$$U^{n+1} = U^n + kg(U^n, t_n) \quad (61a)$$

$$= U^n + k [\lambda U^n + f(U^n, t_n)] \quad (61b)$$

$$= (1 + k\lambda)U^n + kf(U^n, t_n) \quad (61c)$$

As instructed, if we apply the AB1 discretization to the problem,

$$v' = g(v, t),$$

then we have

$$V^{n+1} = V^n + kg(V^n, t_n). \quad (62)$$

Substituting back for u we get

$$e^{-\lambda t_{n+1}} U^{n+1} = e^{-\lambda t_n} U^n + ke^{-\lambda t_n} f(U^n, t_n) \quad (63a)$$

$$U^{n+1} = e^{\lambda(t_{n+1}-t_n)} U^n + ke^{\lambda(t_{n+1}-t_n)} f(U^n, t_n) \quad (63b)$$

$$= e^{\lambda k} U^n + ke^{\lambda k} f(U^n, t_n) \quad (63c)$$

The local truncation error is given by

$$\tau(t_n) = \frac{u(t_n + k) - e^{\lambda k} u(t_n)}{ke^{\lambda k}} - f(u(t_n), t_n) \quad (64a)$$

$$= -\lambda u(t_n) + \frac{1}{2}k (u''(t_n) - 2\lambda u'(t_n) + \lambda^2 u(t_n)) + \mathcal{O}(k^2). \quad (64b)$$

In the limit that $\lambda \rightarrow 0$ we see that the truncation error

$$\lim_{\lambda \rightarrow 0} \tau(t_n) = \lim_{\lambda \rightarrow 0} \left[-\lambda u(t_n) + \frac{1}{2}k (u''(t_n) - 2\lambda u'(t_n) + \lambda^2 u(t_n)) + \mathcal{O}(k^2) \right] \quad (65a)$$

$$= \frac{1}{2}k u''(t_n) + \mathcal{O}(k^2), \quad (65b)$$

which is exactly the truncation error of the FE scheme.

Now applying this exponential time differencing scheme to the test problem,

$$u' = \lambda u,$$

we have that

$$U^{n+1} = e^{\lambda k} U^n + k e^{\lambda k} f(U^n, t_n) \quad (66a)$$

$$= e^{\lambda k} U^n + k e^{\lambda k} [\lambda U^n] \quad (66b)$$

$$= (1 + k\lambda) e^{k\lambda} U^n. \quad (66c)$$

When $\lambda < 0$, we have that $|(1+k\lambda)e^{k\lambda}| \leq 1$ and so we have that this scheme is A-stable.

AB2

Applying the AB2 discretization (54) we have that

$$U^{n+1} = U^n + \frac{k}{2} \{3g(U^n, t_n) - g(U^{n-1}, t_{n-1})\} \quad (67a)$$

$$= U^n + \frac{k}{2} \{3[\lambda U^n + f(U^n, t_n)] - [\lambda U^{n-1} + f(U^n, t_{n-1})]\} \quad (67b)$$

$$= U^n + \frac{k}{2} \{3[\lambda U^n + f(U^n, t_n)] - [\lambda U^{n-1} + f(U^n, t_{n-1})]\} \quad (67c)$$

$$= \frac{1}{2}(2 + 3k\lambda)U^n - \frac{1}{2}k\lambda U^{n-1} + \frac{k}{2} [3f(U^n, t_n) - f(U^{n-1}, t_{n-1})]. \quad (67d)$$

As instructed, if we apply the AB2 discretization to the problem,

$$v' = g(v, t),$$

then we have

$$V^{n+1} = V^n + \frac{k}{2} (3g(V^n, t_n) - g(V^{n-1}, t_{n-1})) \quad (68)$$

Substituting back for u we get

$$e^{-\lambda t_{n+1}} U^{n+1} = e^{-\lambda t_n} U^n + \frac{k}{2} \{3[e^{-\lambda t_n} f(U^n, t_n)] - [e^{-\lambda t_{n-1}} f(U^{n-1}, t_{n-1})]\} \quad (69a)$$

$$U^{n+1} = e^{\lambda k} U^n + \frac{k}{2} \{3e^{\lambda k} f(U^n, t_n) - e^{2\lambda k} f(U^{n-1}, t_{n-1})\}. \quad (69b)$$

Thus we have derived the associated integrating factor scheme for AB2.

RK2

Applying the RK2 discretization (59) we have that

$$U^{n+1} = U^n + \frac{k}{2} [g(U^n, t_n) + g(U^*, t_n + k)], \quad (70)$$

where

$$U^* = (1 + k\lambda)U^n + kf(U^n, t_n),$$

arising from an FE step. Expanding the discretization,

$$U^{n+1} = U^n + \frac{k}{2} \{[\lambda U^n + f(U^n, t_n)] + [\lambda U^* + f(U^*, t_n + k)]\} \quad (71a)$$

$$= U^n + \frac{k}{2} \{\lambda U^n + f(U^n, t_n) + \lambda [(1 + k\lambda)U^n + kf(U^n, t_n)] + f(U^*, t_n + k)\} \quad (71b)$$

$$= \frac{1}{2} [2 + 2k\lambda + (k\lambda)^2] U^n + \frac{k}{2} [(1 + k\lambda)f(U^n, t_n) + f(U^*, t_n + k)] \quad (71c)$$

As instructed, if we apply the RK2 discretization to the problem,

$$v' = g(v, t),$$

then we have

$$V^{n+1} = V^n + \frac{k}{2} [g(V^n, t_n) + g(V^*, t_n + k)] \quad (72)$$

Substituting back for u we get

$$e^{-\lambda t_{n+1}} U^{n+1} = e^{-\lambda t_n} U^n + \{e^{-\lambda t_n} f(U^n, t_n) + [e^{-\lambda t_{n+1}} f(U^*, t_n + k)]\} \quad (73a)$$

$$U^{n+1} = e^{\lambda k} U^n + e^{\lambda k} f(U^n, t_n) + f(U^*, t_n + k). \quad (73b)$$

Therefore, we have derived the associated integrating factor scheme for RK2.

Part C

By integrating factor we showed that

$$u' = \lambda u + f(u, t) \Leftrightarrow (e^{-\lambda t} u)' = e^{-\lambda t} f(u, t). \quad (74)$$

Integrating the RHS of this relation from time t_n to $t_n + k$ we have

$$\int_{t_n}^{t_n+k} dt (e^{-\lambda t} u)' = \int_{t_n}^{t_n+k} dt e^{-\lambda t} f(u, t) \quad (75a)$$

$$u(t_n + k) = e^{\lambda k} u(t_n) + e^{\lambda t_{n+1}} \int_{t_n}^{t_n+k} dt e^{-\lambda t} f(u(t), t). \quad (75b)$$

If we let now $s = t - t_n$ then we can recast the integral bounds and thus reacquire that

$$u(t_n + k) = e^{\lambda k} u(t_n) + e^{\lambda k} \int_0^k ds e^{-\lambda s} f(u(t_n + s), t_n + s). \quad (76)$$

Therefore, we have shown that this integral equation is an equivalent formulation of the differential equation $u' = \lambda u + f(u, t)$, and thus by extension then, any solution u of this differential equation must also be a solution to the equivalent integral equation.

AB1 = FE method

Approximating the integrand by a constant interpolant using the left-hand endpoint of the interval,

$$p(t) = f(u(t_n), t_n) \quad (77)$$

Then we can use this interpolant in the integration,

$$u(t_n + k) \approx e^{\lambda k} u(t_n) + e^{\lambda k} \int_0^k ds e^{-\lambda s} p(s) \quad (78a)$$

$$= e^{\lambda k} u(t_n) + e^{\lambda k} f(u(t_n), t_n) \int_0^k ds e^{-\lambda s} \quad (78b)$$

$$U^{n+1} = e^{\lambda k} U^n + e^{\lambda k} f(U^n, t_n) \left[-\frac{1}{\lambda} e^{-\lambda s} \right]_0^k \quad (78c)$$

$$= e^{\lambda k} U^n + \frac{e^{\lambda k} - 1}{\lambda} f(U^n, t_n). \quad (78d)$$

Which if we add and subtract U^n we can also write this as

$$U^{n+1} = U^n + e^{\lambda k} U^n - U^n + \frac{e^{\lambda k} - 1}{\lambda} f(U^n, t_n) \quad (79a)$$

$$= U^n + \frac{e^{\lambda k} - 1}{\lambda} \lambda U^n + \frac{e^{\lambda k} - 1}{\lambda} f(U^n, t_n) \quad (79b)$$

$$= U^n + \frac{e^{\lambda k} - 1}{\lambda} [\lambda U^n + f(U^n, t_n)] \quad (79c)$$

In the limit as $\lambda \rightarrow 0$ or $k \rightarrow 0$ for fixed λ

$$e^{\lambda k} \approx 1 + \lambda k$$

and so the above expression reduces to

$$U^{n+1} = U^n + \left(\frac{[1 + \lambda k] - 1}{\lambda} \right) [\lambda U^n + f(U^n, t_n)] \quad (80a)$$

$$= (1 + k\lambda) U^n + k f(U^n, t_n), \quad (80b)$$

which is exactly the FE method. Applying this scheme to the test problem

$$u' = \lambda u$$

we acquire

$$U^{n+1} = e^{\lambda k} U^n + \frac{e^{\lambda k} - 1}{\lambda} \lambda U^n \quad (81a)$$

$$= (2e^{\lambda k} - 1) U^n. \quad (81b)$$

For $\lambda < 0 \implies |2e^{\lambda k} - 1| \leq 1$ and so this scheme is A-stable (ie. its stability region contains the entire left-hand plane).

AB2 method

Approximating the integrand by a constant interpolant using the left-hand endpoint of the interval, with $s = t - t_n$,

$$p(s) = \frac{s+k}{k}f^n + \frac{s}{-k}f^{n-1} \quad (82)$$

Then we can use this interpolant in the integration,

$$u(t_n + k) \approx e^{\lambda k}u(t_n) + e^{\lambda k} \int_0^k ds e^{-\lambda s} p(s) \quad (83a)$$

$$= e^{\lambda k}u(t_n) + e^{\lambda k} \int_0^k ds e^{-\lambda s} \left[\frac{s+k}{k}f^n + \frac{s}{-k}f^{n-1} \right] \quad (83b)$$

$$U^{n+1} = e^{\lambda k}U^n + \left[\frac{e^{-\lambda s}(-f^n(k\lambda + \lambda s + 1) + f^{n-1}\lambda s + f^{n-1})}{k\lambda^2} \right]_0^k \quad (83c)$$

$$= e^{\lambda k}U^n + \frac{(k\lambda e^{k\lambda} - 2k\lambda + e^{k\lambda} - 1)}{k\lambda^2}f^n + \frac{(k\lambda - e^{k\lambda} + 1)}{k\lambda^2}f^{n-1} \quad (83d)$$

Thus we have acquired the associated exponential time differencing scheme of the AB2 method. In the limit of $\lambda \rightarrow 0$

$$U^{n+1} = \lim_{\lambda \rightarrow 0} \left\{ [1 + k\lambda] U^n + \frac{(k\lambda[1 + k\lambda] - 2k\lambda + [1 + k\lambda] - 1)}{k\lambda^2} f^n + \frac{(k\lambda - [1 + k\lambda] + 1)}{k\lambda^2} f^{n-1} \right\} \quad (84a)$$

$$= U^n + \frac{k}{2}(3f^n - f^{n-1}), \quad (84b)$$

this scheme reduces to the AB2 method.

RK2 method

Approximating the integrand by a constant interpolant using the left-hand endpoint of the interval, with $s = t - t_n$,

$$p(s) = \frac{t+k}{-k}f^n + \frac{s}{k}f^* \quad (85)$$

Then we can use this interpolant in the integration,

$$u(t_n + k) \approx e^{\lambda k}u(t_n) + e^{\lambda k} \int_0^k ds e^{-\lambda s} p(s) \quad (86a)$$

$$= e^{\lambda k}u(t_n) + e^{\lambda k} \int_0^k ds e^{-\lambda s} \left[\frac{s-k}{-k}f^n + \frac{s}{k}f^* \right] \quad (86b)$$

$$U^{n+1} = e^{\lambda k}U^n + e^{\lambda k} \left[-\frac{e^{-\lambda s}((k\lambda - \lambda s - 1)f^n + (1 + \lambda s)f^*)}{k\lambda^2} \right]_0^k \quad (86c)$$

$$= e^{\lambda k} U^n + \frac{(k\lambda e^{k\lambda} - e^{k\lambda} + 1)}{k\lambda^2} f^n - \frac{(k\lambda - e^{k\lambda} + 1)}{k\lambda^2} f^* \quad (86d)$$

Thus we have obtained the associated exponential time differencing scheme of the RK2 method. In the limit of $\lambda \rightarrow 0$

$$U^{n+1} = \lim_{\lambda \rightarrow 0} \left\{ [1 + k\lambda] U^n + \frac{(k\lambda [1 + k\lambda] - [1 + k\lambda] + 1)}{k\lambda^2} f^n - \frac{(k\lambda - [1 + k\lambda] + 1)}{k\lambda^2} f^* \right\} \quad (87a)$$

$$= U^n + \frac{k}{2} [f^n + f^*], \quad (87b)$$

this scheme reduces to the explicit RK2 method.

Part D

We haven't really talked about this yet. But in this case, the scheme most suited would be the exponential time-differencing. Using this scheme mitigates the stiffness of the equation brought about by the large and negative value of λ . Thus much larger time-steps can be taken, as any large temporal updates to the solution are damped out by the presence of the exponentials in the method.