Math 467/667: Homework 1 Solutions

1. [Iserles 3.1] Find the order of the following quadrature formulae:

(i) $\int_0^1 f(\tau)d\tau = \frac{1}{6}f(0) + \frac{2}{3}f\left(\frac{1}{2}\right) + \frac{1}{6}f(1)$

Recall that a quadrature method is said to be order p if it is exact for polynomials up to degree $p-1$. Therefore, it is enough to compute the integral and quadrature formula for functions $f(x) = x^n$ and report the first value of n for which they differ. For convenience and practice with a computer algebra system we do this using Mathematica.

The Mathematica script

```
qmethod = Function[f, 1/6*f[0] + 2/3*f[1/2] + 1/6*f[1]]
intexact = Function[f, Integrate[f[tau], {tau, 0, 1}]]
For[n = 0, n < 16, n++,
    f = Function[x, Evaluate[x^n]];
    ap = qmethod[f];
    ex = intexact[f];
    Print["n=", n, " ap=", ap, " ex=", ex];
    If[ap - ex == 0, Null, Break[], Break[]]
]
Print["The order is ", n]
```

produces the output

```
Wolfram Language 12.1.1 Engine for Linux ARM (32-bit)
Copyright 1988-2020 Wolfram Research, Inc.

In[1]:= qmethod = Function[f, 1/6*f[0] + 2/3*f[1/2] + 1/6*f[1]]

                                   1
                               2 f[-]
                      1 f[0]       2    1 f[1]
Out[1]= Function[f, ------ + ------ + ------]
                        6        3        6

In[2]:= intexact = Function[f, Integrate[f[tau], {tau, 0, 1}]]

Out[2]= Function[f, Integrate[f[tau], {tau, 0, 1}]]

In[3]:= For[n = 0, n < 16, n++,
            f = Function[x, Evaluate[x^n]];
            ap = qmethod[f];
            ex = intexact[f];
            Print["n=", n, " ap=", ap, " ex=", ex];
            If[ap - ex == 0, Null, Break[], Break[]]
        ]
n=0 ap=1 ex=1
```

```
       1    1
n=1 ap=- ex=-
       2    2
       1    1
n=2 ap=- ex=-
       3    3
       1    1
n=3 ap=- ex=-
       4    4
       5     1
n=4 ap=-- ex=-
       24    5

In[4]:= Print["The order is ", n]
The order is 4

In[5]:=
```

Therefore the method is order 4.

(ii) $\int_0^1 f(\tau)d\tau = \frac{1}{8}f(0) + \frac{3}{8}f\left(\frac{1}{3}\right) + \frac{3}{8}f\left(\frac{2}{23}\right) + \frac{1}{8}f(1)$

The Mathematica script

```
qmethod = Function[f, 1/8*f[0] + 3/8f[1/3] + 3/8*f[2/3] + 1/8*f[1]]
intexact = Function[f, Integrate[f[tau], {tau, 0, 1}]]
For[n = 0, n < 16, n++,
    f = Function[x, Evaluate[x^n]];
    ap = qmethod[f];
    ex = intexact[f];
    Print["n=", n, " ap=", ap, " ex=", ex];
    If[ap - ex == 0, Null, Break[], Break[]]
]
Print["The order is ", n]
```

produces the output

```
Wolfram Language 12.1.1 Engine for Linux ARM (32-bit)
Copyright 1988-2020 Wolfram Research, Inc.

In[1]:= qmethod = Function[f, 1/8*f[0] + 3/8f[1/3] + 3/8*f[2/3] + 1/8*f[1]]

                                  1        2
                              3 f[-]   3 f[-]
                     1 f[0]       3        3    1 f[1]
Out[1]= Function[f, ------ + ------ + ------ + ------]
                       8        8        8        8
```

```
In[2]:= intexact = Function[f, Integrate[f[tau], {tau, 0, 1}]]

Out[2]= Function[f, Integrate[f[tau], {tau, 0, 1}]]

In[3]:= For[n = 0, n < 16, n++,
            f = Function[x, Evaluate[x^n]];
            ap = qmethod[f];
            ex = intexact[f];
            Print["n=", n, " ap=", ap, " ex=", ex];
            If[ap - ex == 0, Null, Break[], Break[]]
        ]
n=0 ap=1 ex=1
       1    1
n=1 ap=- ex=-
       2    2
       1    1
n=2 ap=- ex=-
       3    3
       1    1
n=3 ap=- ex=-
       4    4
       11    1
n=4 ap=-- ex=-
       54    5

In[4]:= Print["The order is ", n]
The order is 4

In[5]:=
```

Therefore the method is order 4.

(iii) $\displaystyle\int_0^1 f(\tau)d\tau = \tfrac{2}{3}f\left(\tfrac{1}{4}\right) - \tfrac{1}{3}f\left(\tfrac{1}{2}\right) + \tfrac{1}{6}f\left(\tfrac{3}{4}\right)$

The Mathematica script

```
qmethod = Function[f, 2/3*f[1/4] - 1/3*f[1/2] + 2/3*f[3/4]]
intexact = Function[f, Integrate[f[tau], {tau, 0, 1}]]
For[n = 0, n < 16, n++,
    f = Function[x, Evaluate[x^n]];
    ap = qmethod[f];
    ex = intexact[f];
    Print["n=", n, " ap=", ap, " ex=", ex];
    If[ap - ex == 0, Null, Break[], Break[]]
]
```

```
Print["The order is ", n]
```

produces the output

```
Wolfram Language 12.1.1 Engine for Linux ARM (32-bit)
Copyright 1988-2020 Wolfram Research, Inc.

In[1]:= qmethod = Function[f, 2/3*f[1/4] - 1/3*f[1/2] + 2/3*f[3/4]]

                        1        1        3
                    2 f[-]   1 f[-]   2 f[-]
                        4        2        4
Out[1]= Function[f, ------ - ------ + ------]
                      3        3        3

In[2]:= intexact = Function[f, Integrate[f[tau], {tau, 0, 1}]]

Out[2]= Function[f, Integrate[f[tau], {tau, 0, 1}]]

In[3]:= For[n = 0, n < 16, n++,
            f = Function[x, Evaluate[x^n]];
            ap = qmethod[f];
            ex = intexact[f];
            Print["n=", n, " ap=", ap, " ex=", ex];
            If[ap - ex == 0, Null, Break[], Break[]]
        ]
n=0 ap=1 ex=1
       1    1
n=1 ap=- ex=-
       2    2
       1    1
n=2 ap=- ex=-
       3    3
       1    1
n=3 ap=- ex=-
       4    4
       37     1
n=4 ap=--- ex=-
       192    5

In[4]:= Print["The order is ", n]
The order is 4

In[5]:=
```

Therefore the method is order 4.

(iv) $\displaystyle\int_0^\infty f(\tau)e^{-\tau}d\tau = \frac{5}{3}f(1) - \frac{3}{2}f(2) + f(3) - \frac{1}{6}f(4)$

The Mathematica script

```
qmethod = Function[f, 5/3*f[1] - 3/2*f[2] + f[3] - 1/6*f[4]]
intexact = Function[f, Integrate[f[tau]*Exp[-tau], {tau, 0, Infinity}]]
For[n = 0, n < 16, n++,
    f = Function[x, Evaluate[x^n]];
    ap = qmethod[f];
    ex = intexact[f];
    Print["n=", n, " ap=", ap, " ex=", ex];
    If[ap - ex == 0, Null, Break[], Break[]]
]
Print["The order is ", n]
```

produces the output

```
Wolfram Language 12.1.1 Engine for Linux ARM (32-bit)
Copyright 1988-2020 Wolfram Research, Inc.

In[1]:= qmethod = Function[f, 5/3*f[1] - 3/2*f[2] + f[3] - 1/6*f[4]]

                    5 f[1]   3 f[2]          1 f[4]
Out[1]= Function[f, ------ - ------ + f[3] - ------]
                      3        2               6

In[2]:= intexact = Function[f, Integrate[f[tau]*Exp[-tau], {tau, 0, Infinity}]]

Out[2]= Function[f, Integrate[f[tau] Exp[-tau], {tau, 0, Infinity}]]

In[3]:= For[n = 0, n < 16, n++,
            f = Function[x, Evaluate[x^n]];
            ap = qmethod[f];
            ex = intexact[f];
            Print["n=", n, " ap=", ap, " ex=", ex];
            If[ap - ex == 0, Null, Break[], Break[]]
        ]
n=0 ap=1 ex=1
n=1 ap=1 ex=1
n=2 ap=2 ex=2
n=3 ap=6 ex=6
n=4 ap=16 ex=24

In[4]:= Print["The order is ", n]
The order is 4
```

```
In[5]:=
```

Therefore the method is order 4.

2. [Iserles 3.6] Determine all choices of A, b and c such that the resulting two-stage IRK method is order $p \geq 3$.

The general form of a two-stage IRK method is

$$\begin{aligned} k_1 &= f\big(t_n + hc_1, y_n + h(a_{11}k_1 + a_{12}k_2)\big) \\ k_2 &= f\big(t_n + hc_2, y_n + h(a_{21}k_1 + a_{22}k_2)\big) \\ y_{n+1} &= y_n + h(b_1 k_1 + b_2 k_2) \end{aligned}$$

where for consistency we already know that

$$c_1 = a_{11} + a_{12}, \qquad c_2 = a_{21} + a_{22} \qquad \text{and} \qquad 1 = b_1 + b_2.$$

Eliminating c_1, c_2 and b_2 leaves five variables that may be used to satisfy the required order conditions. To this end we consider the truncation error

$$\tau(h) = y(t+h) - y(t) - h(b_1 k_1 + b_2 k_2)$$

where k_1 and k_2 are viewed as implicit functions of h and t such that

$$\begin{aligned} k_1 &= f\big(t + hc_1, y(t) + h(a_{11}k_1 + a_{12}k_2)\big) \\ k_2 &= f\big(t + hc_2, y(t) + h(a_{21}k_1 + a_{22}k_2)\big). \end{aligned}$$

Now, expand τ about $h = 0$ as

$$\tau(h) = \tau(0) + h\tau'(0) + \frac{h^2}{2!}\tau''(0) + \frac{h^3}{3!}\tau^{(3)}(0) + \mathcal{O}(h^4)$$

and then find conditions on the remaining parameters such that

$$\tau(0) = 0, \qquad \tau'(0) = 0, \qquad \tau''(0) = 0 \qquad \text{and} \qquad \tau^{(3)}(0) = 0.$$

This we do using a computer algebra system as follows:

To ensure $\tau''(0) = 0$ we obtain the condition

$$1 + 2a_{21}(-1 + b1) + 2a_{22}(-1 + b_1) - 2a_{11}b_1 - 2a_{12}b1 = 0$$

and use this to eliminate b_1 as

$$b_1 = \frac{-1 + 2a_{21} + 2a_{22}}{2(a_{11} + a_{12} - a_{21} - a_{22})}$$

To ensure $\tau'''(0) = 0$ we first obtain the condition

$$2 - 3a_{21} - 3a_{22} + a_{11}(-3 + 6a_{21} + 6a_{22}) + a_{12}(-3 + 6a_{21} + 6a_{22}) = 0$$

and use this to eliminate a_{12} as

$$a_{12} = \frac{-2 + 3a_{11} + 3a_{21} - 6a_{11}a_{21} + 3a_{22} - 6a_{11}a_{22}}{3(-1 + 2a_{21} + 2a_{22})}.$$

This leads to another condition

$$-1 - 6a_{21}^2 + a_{21}(6 - 12a_{22}) + 5a_{22} - 6a_{22}^2 + 3a_{11}(-1 + 2a_{21} + 2a_{22})^2 = 0$$

which we then use to eliminate a_{11} as

$$a_{11} = \frac{1 + 6a_{21}^2 - 5a_{22} + 6a_{22}^2 + 6a_{21}(-1 + 2a_{22})}{3(-1 + 2a_{21} + 2a_{22})^2.}$$

This leaves two variables a_{21} and a_{22} that corresponding to a two-parameter family of IRK methods of order $p \geq 3$. Note that one could eliminate other variables in favor of the ones used here. This would lead to the same two-parameter family of IRK methods, just parameterized in a different way.

For reference, the Mathematica script used for the above calculations was

```
ode=y'->Function[t,f[t,y[t]]]
eq1=k1[h]==f[t+c1*h,y[t]+h*(a11*k1[h]+a12*k2[h])]
eq2=k2[h]==f[t+c2*h,y[t]+h*(a21*k1[h]+a22*k2[h])]
r=y[t+h]-y[t]-h*(b1*k1[h]+b2*k2[h])
c1=a11+a12
c2=a21+a22
b2=1-b1
ode
eq1
eq2
r
deq1=D[eq1,h]/.ode
deq2=D[eq2,h]/.ode
dk1=Solve[{deq1,deq2},{k1'[h],k2'[h]}][[1]][[1]];
dk2=Solve[{deq1,deq2},{k1'[h],k2'[h]}][[1]][[2]];
sk1=k1[0]->f[t,y[t]]
sk2=k2[0]->f[t,y[t]]
dr=D[r,h]/.{ode,dk1,dk2};
Simplify[dr/.h->0/.{sk1,sk2}]
d2r=D[dr,h]/.{ode,dk1,dk2};
d2r0=Simplify[d2r/.h->0/.{sk1,sk2}]
cond4=Coefficient[d2r0,Derivative[1,0][f][t,y[t]]]==0
```

```
sub4=Solve[cond4,b1][[1]][[1]]
newb1=b1/.sub4
b1=newb1
Simplify[d2r0]
d3r=D[d2r,h]/.{ode,dk1,dk2};
d3r0=Simplify[d3r/.h->0/.{sk1,sk2}]
cond5=Simplify[Coefficient[d3r0,Derivative[2,0][f][t,y[t]]]]==0
sub5=Solve[cond5,a12][[1]][[1]]
newa12=a12/.sub5
a12=newa12
d3r00=Simplify[d3r0]
cond6=Coefficient[
    Coefficient[d3r00,Derivative[0,1][f][t,y[t]]],
    Derivative[1,0][f][t,y[t]]]==0
sub6=Solve[cond6,a11][[1]][[1]]
newa11=Simplify[a11/.sub6]
a11=newa11
Simplify[d3r00]
```

and the resulting output

```
Wolfram Language 12.1.1 Engine for Linux ARM (32-bit)
Copyright 1988-2020 Wolfram Research, Inc.

In[1]:= ode=y'->Function[t,f[t,y[t]]]

Out[1]= y' -> Function[t, f[t, y[t]]]

In[2]:= eq1=k1[h]==f[t+c1*h,y[t]+h*(a11*k1[h]+a12*k2[h])]

Out[2]= k1[h] == f[c1 h + t, h (a11 k1[h] + a12 k2[h]) + y[t]]

In[3]:= eq2=k2[h]==f[t+c2*h,y[t]+h*(a21*k1[h]+a22*k2[h])]

Out[3]= k2[h] == f[c2 h + t, h (a21 k1[h] + a22 k2[h]) + y[t]]

In[4]:= r=y[t+h]-y[t]-h*(b1*k1[h]+b2*k2[h])

Out[4]= -(h (b1 k1[h] + b2 k2[h])) - y[t] + y[h + t]

In[5]:= c1=a11+a12

Out[5]= a11 + a12

In[6]:= c2=a21+a22
```

```
Out[6]= a21 + a22

In[7]:= b2=1-b1

Out[7]= 1 - b1

In[8]:= ode

Out[8]= y' -> Function[t, f[t, y[t]]]

In[9]:= eq1

Out[9]= k1[h] == f[(a11 + a12) h + t, h (a11 k1[h] + a12 k2[h]) + y[t]]

In[10]:= eq2

Out[10]= k2[h] == f[(a21 + a22) h + t, h (a21 k1[h] + a22 k2[h]) + y[t]]

In[11]:= r

Out[11]= -(h (b1 k1[h] + (1 - b1) k2[h])) - y[t] + y[h + t]

In[12]:= deq1=D[eq1,h]/.ode

Out[12]= k1'[h] == (a11 k1[h] + a12 k2[h] + h (a11 k1'[h] + a12 k2'[h]))

        (0,1)
>      f     [(a11 + a12) h + t, h (a11 k1[h] + a12 k2[h]) + y[t]] +

                   (1,0)
>     (a11 + a12) f     [(a11 + a12) h + t, h (a11 k1[h] + a12 k2[h]) + y[t]]

In[13]:= deq2=D[eq2,h]/.ode

Out[13]= k2'[h] == (a21 k1[h] + a22 k2[h] + h (a21 k1'[h] + a22 k2'[h]))

        (0,1)
>      f     [(a21 + a22) h + t, h (a21 k1[h] + a22 k2[h]) + y[t]] +

                   (1,0)
>     (a21 + a22) f     [(a21 + a22) h + t, h (a21 k1[h] + a22 k2[h]) + y[t]]

In[14]:= dk1=Solve[{deq1,deq2},{k1'[h],k2'[h]}][[1]][[1]];
```

```
In[15]:= dk2=Solve[{deq1,deq2},{k1'[h],k2'[h]}][[1]][[2]];

In[16]:= sk1=k1[0]->f[t,y[t]]

Out[16]= k1[0] -> f[t, y[t]]

In[17]:= sk2=k2[0]->f[t,y[t]]

Out[17]= k2[0] -> f[t, y[t]]

In[18]:= dr=D[r,h]/.{ode,dk1,dk2};

In[19]:= Simplify[dr/.h->0/.{sk1,sk2}]

Out[19]= 0

In[20]:= d2r=D[dr,h]/.{ode,dk1,dk2};

In[21]:= d2r0=Simplify[d2r/.h->0/.{sk1,sk2}]

Out[21]= (1 + 2 a21 (-1 + b1) + 2 a22 (-1 + b1) - 2 a11 b1 - 2 a12 b1)

                  (0,1)             (1,0)
>    (f[t, y[t]] f     [t, y[t]] + f     [t, y[t]])

In[22]:= cond4=Coefficient[d2r0,Derivative[1,0][f][t,y[t]]]==0

Out[22]= 1 + 2 a21 (-1 + b1) + 2 a22 (-1 + b1) - 2 a11 b1 - 2 a12 b1 == 0

In[23]:= sub4=Solve[cond4,b1][[1]][[1]]

                 -(-1 + 2 a21 + 2 a22)
Out[23]= b1 -> -------------------------
               2 (a11 + a12 - a21 - a22)

In[24]:= newb1=b1/.sub4

           -(-1 + 2 a21 + 2 a22)
Out[24]= -------------------------
         2 (a11 + a12 - a21 - a22)

In[25]:= b1=newb1
```

```
           -(-1 + 2 a21 + 2 a22)
Out[25]= -------------------------
         2 (a11 + a12 - a21 - a22)

In[26]:= Simplify[d2r0]

Out[26]= 0

In[27]:= d3r=D[d2r,h]/.{ode,dk1,dk2};

In[28]:= d3r0=Simplify[d3r/.h->0/.{sk1,sk2}]

Out[28]= ((2 - 3 a21 - 3 a22 + a11 (-3 + 6 a21 + 6 a22) +

                                                    2  (0,2)
>         a12 (-3 + 6 a21 + 6 a22)) f[t, y[t]]  f     [t, y[t]]) / 2 +

                                                       (0,1)
>    (1 - 6 a12 a21 - 3 a22 + a11 (-3 + 6 a22)) f     [t, y[t]]

       (1,0)
>     f     [t, y[t]] + f[t, y[t]]

                                                         (0,1)         2
>     ((1 - 6 a12 a21 - 3 a22 + a11 (-3 + 6 a22)) f     [t, y[t]]  +

>       (2 - 3 a21 - 3 a22 + a11 (-3 + 6 a21 + 6 a22) +

                                            (1,1)
>          a12 (-3 + 6 a21 + 6 a22)) f     [t, y[t]]) +

>    ((2 - 3 a21 - 3 a22 + a11 (-3 + 6 a21 + 6 a22) + a12 (-3 + 6 a21 + 6 a22))

         (2,0)
>       f     [t, y[t]]) / 2

In[29]:= cond5=Simplify[Coefficient[d3r0,Derivative[2,0][f][t,y[t]]]]==0

Out[29]=

      2 - 3 a21 - 3 a22 + a11 (-3 + 6 a21 + 6 a22) + a12 (-3 + 6 a21 + 6 a22)
>     ----------------------------------------------------------------------\
                                      2
```

```
>     == 0

In[30]:= sub5=Solve[cond5,a12][[1]][[1]]

                -2 + 3 a11 + 3 a21 - 6 a11 a21 + 3 a22 - 6 a11 a22
Out[30]= a12 -> --------------------------------------------------
                              3 (-1 + 2 a21 + 2 a22)

In[31]:= newa12=a12/.sub5

         -2 + 3 a11 + 3 a21 - 6 a11 a21 + 3 a22 - 6 a11 a22
Out[31]= --------------------------------------------------
                       3 (-1 + 2 a21 + 2 a22)

In[32]:= a12=newa12

         -2 + 3 a11 + 3 a21 - 6 a11 a21 + 3 a22 - 6 a11 a22
Out[32]= --------------------------------------------------
                       3 (-1 + 2 a21 + 2 a22)

In[33]:= d3r00=Simplify[d3r0]

                      2                                    2
Out[33]= ((-1 - 6 a21  + a21 (6 - 12 a22) + 5 a22 - 6 a22  +

                                            2   (0,1)
>        3 a11 (-1 + 2 a21 + 2 a22) ) f     [t, y[t]]

                    (0,1)              (1,0)
>      (f[t, y[t]] f     [t, y[t]] + f     [t, y[t]])) / (-1 + 2 a21 + 2 a22)

In[34]:= cond6=Coefficient[
             Coefficient[d3r00,Derivative[0,1][f][t,y[t]]],
             Derivative[1,0][f][t,y[t]]]==0

                    2                                    2
Out[34]= (-1 - 6 a21  + a21 (6 - 12 a22) + 5 a22 - 6 a22  +

                                           2
>       3 a11 (-1 + 2 a21 + 2 a22) ) / (-1 + 2 a21 + 2 a22) == 0

In[35]:= sub6=Solve[cond6,a11][[1]][[1]]
```

```
                         1                  6 a21           a21 (6 - 12 a22)
Out[35]= a11 -> (----------------- + ----------------- - ----------------- - 
                 -1 + 2 a21 + 2 a22   -1 + 2 a21 + 2 a22   -1 + 2 a21 + 2 a22

                                                2
              5 a22               6 a22
>       ----------------- + -----------------) / (3 (-1 + 2 a21 + 2 a22))
        -1 + 2 a21 + 2 a22   -1 + 2 a21 + 2 a22

In[36]:= newa11=Simplify[a11/.sub6]

                  2                2
         1 + 6 a21  - 5 a22 + 6 a22  + 6 a21 (-1 + 2 a22)
Out[36]= ------------------------------------------------
                                              2
                     3 (-1 + 2 a21 + 2 a22)

In[37]:= a11=newa11

                  2                2
         1 + 6 a21  - 5 a22 + 6 a22  + 6 a21 (-1 + 2 a22)
Out[37]= ------------------------------------------------
                                              2
                     3 (-1 + 2 a21 + 2 a22)

In[38]:= Simplify[d3r00]

Out[38]= 0

In[39]:=
```

3. [Iserles 3.7] Write the theta method

$$y_{n+1} = y_n + h\big[\theta f(t_n, y_n) + (1-\theta) f(t_{n+1}, y_{n+1})\big]$$

as a Runge–Kutta method.

Applying $f(t_{n+1}, \cdot)$ to both sides of the theta method yields

$$f(t_{n+1}, y_{n+1}) = f\big(t_{n+1}, y_n + h\big[\theta f(t_n, y_n) + (1-\theta) f(t_{n+1}, y_{n+1})\big]\big)$$

Now writing

$$k_1 = f(t_n, y_n) \qquad \text{and} \qquad k_2 = f(t_{n+1}, y_{n+1})$$

yields the IRK method

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f\big(t_n + h, y_n + h(\theta k_1 + (1-\theta) k_2)\big) \\ y_{n+1} &= y_n + h\big(\theta k_1 + (1-\theta) k_2\big). \end{aligned}$$

In tableau form this scheme may be expressed

$$\begin{array}{c|cc} 0 & & \\ 1 & \theta & 1-\theta \\ \hline & \theta & 1-\theta \end{array}.$$

4. Complete the multi-part question about Gaussian quadrature:

(i) Make the change of variables $y = \tan z$ so that

$$\int_0^{\pi/2} \frac{dz}{\sqrt{1+\tan z}} = \int_0^\infty g(y)dy.$$

Write down an explicit formula for $g(y)$.

Since $y = \tan z$ it follows that $z = \arctan y$. Consequently

$$dz = \frac{dy}{1+y^2}, \qquad \lim_{z\nearrow\pi/2} \tan z = \infty \qquad \text{and} \qquad \tan 0 = 0$$

imply

$$\int_0^{\pi/2} \frac{dz}{\sqrt{1+\tan z}} = \int_0^\infty \frac{1}{\sqrt{1+y}} \cdot \frac{dy}{1+y^2}.$$

It follows that

$$g(y) = \frac{1}{(1+y^2)\sqrt{1+y}}.$$

(ii) Show the further change of variables

$$x = \frac{2y}{1+y} - 1$$

transforms the integral above into the form

$$\int_0^\infty g(y)dy = \int_{-1}^1 h(x)\sqrt{1-x}\,dx.$$

Write down an explicit formula for $h(x)$.

Solving for y in terms of x yields

$$1+x = \frac{2y}{1+y}, \qquad (1+x)(1+y) = 2y \qquad \text{so} \qquad y = \frac{1+x}{1-x}.$$

Therefore

$$dy = \frac{2dx}{(1-x)^2}, \qquad \lim_{y\to\infty}\Big(\frac{2y}{1+y} - 1\Big) = 1 \qquad \text{and} \qquad \frac{2\cdot 0}{1+0} - 1 = -1$$

imply

$$\int_0^\infty g(y)dy = \int_{-1}^1 g\Big(\frac{1+x}{1-x}\Big)\cdot\frac{2dx}{(1-x)^2}.$$

Since

$$g\Big(\frac{1+x}{1-x}\Big) = \frac{1}{\Big(1+\big(\frac{1+x}{1-x}\big)^2\Big)\sqrt{1+\frac{1+x}{1-x}}}$$

$$= \frac{(1-x)^2}{(1-x)^2+(1+x)^2}\cdot\frac{\sqrt{1-x}}{\sqrt{1-x+1+x}} = \frac{(1-x)^2\sqrt{1-x}}{2\sqrt{2}(1+x^2)},$$

we obtain

$$\int_0^\infty g(y)dy = \int_{-1}^1 \frac{\sqrt{1-x}}{\sqrt{2}(1+x^2)}dx.$$

Consequently,

$$h(x) = \frac{1}{\sqrt{2}(1+x^2)}.$$

(iii) Define the weighted inner product and norm as

$$(\alpha,\beta) = \int_{-1}^1 \alpha(x)\beta(x)\sqrt{1-x}\,dx \qquad \text{and} \qquad \|\alpha\| = \sqrt{(\alpha,\alpha)}$$

Find the orthogonal polynomials p_n of degree n with respect to this inner product for $n = 0, 1, \ldots, 6$.

The Mathematica script

```
dp = Function[{p, q}, Integrate[p*q*Sqrt[1-x], {x, -1, 1}]]
nm = Function[p, Sqrt[dp[p, p]]]
n = 6
For[k = 0, k <= n, k++,
    u[k] = x^k
]
For[j = 0, j <= n, j++,
    v[j] = u[j];
    For[k = 0, k < j, k++,
        v[j] = v[j] - dp[u[j], p[k]]*p[k]
    ];
    p[j] = Simplify[v[j]/nm[v[j]]]
]
For[j = 0, j <= n, j++,
    Print["p[", j, "] = ", p[j]]
]
```

performs Gram–Schmidt orthogonalization to obtain the desired polynomials as

```
Wolfram Language 12.1.1 Engine for Linux ARM (32-bit)
Copyright 1988-2020 Wolfram Research, Inc.

In[1]:= dp = Function[{p, q}, Integrate[p*q*Sqrt[1-x], {x, -1, 1}]]
```

```
Out[1]= Function[{p, q}, Integrate[p q Sqrt[1 - x], {x, -1, 1}]]

In[2]:= nm = Function[p, Sqrt[dp[p, p]]]

Out[2]= Function[p, Sqrt[dp[p, p]]]

In[3]:= n = 6

Out[3]= 6

In[4]:= For[k = 0, k <= n, k++,
            u[k] = x^k
        ]

In[5]:= For[j = 0, j <= n, j++,
            v[j] = u[j];
            For[k = 0, k < j, k++,
                v[j] = v[j] - dp[u[j], p[k]]*p[k]
            ];
            p[j] = Simplify[v[j]/nm[v[j]]]
        ]

In[6]:= For[j = 0, j <= n, j++,
            Print["p[", j, "] = ", p[j]]
        ]
       Sqrt[3]
p[0] = -------
          1/4
       2 2
       Sqrt[7] (1 + 5 x)
p[1] = -----------------
               1/4
            8 2
                                     2
       Sqrt[11] (-17 + 14 x + 63 x )
p[2] = -----------------------------
                      1/4
                  64 2
                                      2        3
       Sqrt[15] (-23 - 225 x + 99 x  + 429 x )
p[3] = ----------------------------------------
                           1/4
                      256 2
```

```
                                           2         3          4
       Sqrt[19] (827 - 1364 x - 9438 x  + 2860 x  + 12155 x )
p[4] = ------------------------------------------------------
                                        1/4
                                  4096 2
                                              2          3          4          5
       Sqrt[23] (1207 + 17615 x - 15210 x  - 90610 x  + 20995 x  + 88179 x )
p[5] = ---------------------------------------------------------------------
                                                1/4
                                         16384 2
                                                   2           3            4
p[6] = (3 Sqrt[3] (-22181 + 54930 x + 512805 x  - 303620 x  - 1661835 x  +

                 5            6              1/4
>        312018 x  + 1300075 x )) / (131072 2   )

In[7]:=
```

(iv) Find the six roots x_k of p_6 and the corresponding weights w_k for $k = 1, 2, \ldots, 6$ such that

$$\int_{-1}^{1} x^j \sqrt{1-x}\, dx = \sum_{k=1}^{6} w_k x_k^j \qquad \text{for} \qquad j = 0, 1, \ldots, 11.$$

Modifications of the previous Mathematica script

```
dp = Function[{p, q}, Integrate[p*q*Sqrt[1-x], {x, -1, 1}]];
nm = Function[p, Sqrt[dp[p, p]]];
n = 6;
For[k = 0, k <= n, k++,
    u[k] = x^k
]
For[j = 0, j <= n, j++,
    v[j] = u[j];
    For[k = 0, k < j, k++,
        v[j] = v[j] - dp[u[j], p[k]]*p[k]
    ];
    p[j] = Simplify[v[j]/nm[v[j]]]
]
xs=Solve[p[n]==0,x];

Print["X = ["]; For[k = 1, k <= n, k++,
    Print["    ",N[x/.xs[[k]],17]]
]; Print["]"]

```

```
Print["B = ["]; For[j = 0, j <= 11, j++,
    b = Integrate[x^j*Sqrt[1-x],{x,-1,1}];
    Print["    ",N[b,17]]
]; Print["]"]
```

to obtain the roots of p_6 as well as the values

$$b_j = \int_{-1}^{1} x^j \sqrt{1-x}\, dx$$

yields

```
Wolfram Language 12.1.1 Engine for Linux ARM (32-bit)
Copyright 1988-2020 Wolfram Research, Inc.

In[1]:= dp = Function[{p, q}, Integrate[p*q*Sqrt[1-x], {x, -1, 1}]];

In[2]:= nm = Function[p, Sqrt[dp[p, p]]];

In[3]:= n = 6;

In[4]:= For[k = 0, k <= n, k++,
            u[k] = x^k
        ]

In[5]:= For[j = 0, j <= n, j++,
            v[j] = u[j];
            For[k = 0, k < j, k++,
                v[j] = v[j] - dp[u[j], p[k]]*p[k]
            ];
            p[j] = Simplify[v[j]/nm[v[j]]]
        ]

In[6]:= xs=Solve[p[n]==0,x];

In[7]:=
In[7]:= Print["X = ["]; For[k = 1, k <= n, k++,
            Print["    ",N[x/.xs[[k]],17]]
        ]; Print["]"]
X = [
    -0.93723257039042295
    -0.68397364451314330
    -0.28505487108731684
    0.17477465224111716
    0.59770850453551940
    0.89377792921424653
```

```
]

In[8]:=
In[8]:= Print["B = ["]; For[j = 0, j <= 11, j++,
            b = Integrate[x^j*Sqrt[1-x],{x,-1,1}];
            Print["    ",N[b,17]]
        ]; Print["]"]
B = [
    1.8856180831641267
    -0.37712361663282535
    0.59262282613729697
    -0.23345747696317760
    0.34447222125335995
    -0.17016400284298313
    0.24099241435843884
    -0.13429179108671979
    0.18464134695288756
    -0.11111000020668591
    0.14933279327646356
    -0.094861311896407273
]

In[9]:=
```

The corresponding values of x_k and b_j are then plugged into Julia to obtain the weights w_k along with a verification of the accuracy of the method. In particular, the program

```
using Printf

X = [ -0.93723257039042295, -0.68397364451314330, -0.28505487108731684,
    0.17477465224111716, 0.59770850453551940, 0.89377792921424653 ]
B = [ 1.8856180831641267, -0.37712361663282535, 0.59262282613729697,
    -0.23345747696317760, 0.34447222125335995, -0.17016400284298313,
    0.24099241435843884, -0.13429179108671979, 0.18464134695288756,
    -0.11111000020668591, 0.14933279327646356, -0.094861311896407273 ]
V=X'.^[0:6;]
W=V\B[1:7]

@printf("%4s %22s\n","k","W[k]")
for k=1:6
    @printf("%4d %22.14e\n",k,W[k])
end
@printf("\n%4s %22s %22s %22s\n",
    "j","quad(x^j)","int(x^j)","error")
for j=0:11
```

```
    approx=W'*(x->x^j).(X)
    error=approx-B[j+1]
    @printf("%4d %22.14e %22.14e %22.14e\n",
        j,approx,B[j+1],error)
end
```

produces the output

```
   k                    W[k]
   1   2.21824863508191e-01
   2   4.38774402812100e-01
   3   5.04761333142555e-01
   4   4.15808709059290e-01
   5   2.36463942277132e-01
   6   6.79848323648581e-02

   j             quad(x^j)              int(x^j)                 error
   0   1.88561808316413e+00   1.88561808316413e+00   0.00000000000000e+00
   1  -3.77123616632825e-01  -3.77123616632825e-01  -1.11022302462516e-16
   2   5.92622826137297e-01   5.92622826137297e-01   1.11022302462516e-16
   3  -2.33457476963178e-01  -2.33457476963178e-01  -1.94289029309402e-16
   4   3.44472221253360e-01   3.44472221253360e-01   1.11022302462516e-16
   5  -1.70164002842983e-01  -1.70164002842983e-01  -2.77555756156289e-17
   6   2.40992414358439e-01   2.40992414358439e-01   8.32667268468867e-17
   7  -1.34291791086720e-01  -1.34291791086720e-01   2.77555756156289e-17
   8   1.84641346952888e-01   1.84641346952888e-01   2.77555756156289e-17
   9  -1.11110000206686e-01  -1.11110000206686e-01   2.77555756156289e-17
  10   1.49332793276464e-01   1.49332793276464e-01   2.77555756156289e-17
  11  -9.48613118964072e-02  -9.48613118964073e-02   4.16333634234434e-17
```

This shows, up to rounding error, that the equality

$$\int_{-1}^{1} p(x)\sqrt{1-x}\,dx = \sum_{k=1}^{6} w_k x_k^j$$

is exact for all polynomials p of degree 11 or less.

(v) Use the weighted six-point Gauss quadrature method and the change of variables developed above to approximate the integral

$$\int_0^{\pi/2} \frac{dz}{\sqrt{1+\tan z}} \approx \sum_{k=1}^{6} w_k x_k^j.$$

What is the error in the approximation? Hint: if it's way off, please check all of your work and fix the mistake.

First compute the exact value using Mathematica as

Math 467/667: Homework 1 Solutions

```
Wolfram Language 12.1.1 Engine for Linux ARM (32-bit)
Copyright 1988-2020 Wolfram Research, Inc.

In[1]:= r1 = Integrate[1/Sqrt[1 + Tan[z]], {z, 0, Pi/2}]

            2 Pi          2 Pi                3/2              1
Out[1]= (----------- + ----------- - 2 (1 - I)    ArcTanh[-----------] -
         Sqrt[1 - I]   Sqrt[1 + I]                         Sqrt[1 - I]

                3/2             1
>      2 (1 + I)    ArcTanh[-----------]) / 4
                            Sqrt[1 + I]

In[2]:= Print["exact=", Re[N[r1, 17]]]
exact=1.06023329227074372

In[3]:=
```

Copy the exact value of the integral into Julia and modify the previous program to obtain

```julia
using Printf

X = [ -0.93723257039042295, -0.68397364451314330, -0.28505487108731684,
    0.17477465224111716, 0.59770850453551940, 0.89377792921424653 ]
B = [ 1.8856180831641267, -0.37712361663282535, 0.59262282613729697,
    -0.23345747696317760, 0.34447222125335995, -0.17016400284298313,
    0.24099241435843884 ]
V=X'.^[0:6;]
W=V\B
h(x)=1/(sqrt(2)*(1+x^2))
exact=1.06023329227074372
approx=W'*h.(X)
error=approx-exact
@printf("%22s %22s %22s\n",
    "approx","exact","error")
@printf("%22.14e %22.14e %22.14e\n",
    approx,exact,error)
```

The resulting computation

```
                approx                  exact                  error
  1.06020035234897e+00   1.06023329227074e+00  -3.29399217706694e-05
```

computes the error and shows the approximation is good to 5 significant digits.

5. [Iserles 4.4] Determine all values of θ such that the theta method

$$y_{n+1} = y_n + h\big[\theta f(t_n, y_n) + (1-\theta) f(t_{n+1}, y_{n+1})\big]$$

is A-stable.

As this is a simple method, compute the linear stability domain by hand. To do this consider $y' = \lambda y$ with respect to which the theta method becomes This finite difference has a solution of the form $y_n = \omega^n$. Substituting in y_n and writing $z = \lambda h$ yields

$$\omega^{n+1} = \omega^n + z\big[\theta\omega^n + (1-\theta)\omega^{n+1}\big].$$

consequently

$$\omega(1 - z(1-\theta)) = 1 + z\theta \qquad \text{so that} \qquad \omega = \frac{1+z\theta}{1-z(1-\theta)}.$$

By definition the linear stability domain is

$$\mathcal{D}_\theta = \Big\{\, z \in \mathbf{C} : \Big|\frac{1+z\theta}{1-z(1-\theta)}\Big| < 1 \,\Big\}$$

and to be A-stable means $\mathbf{C}^- \subseteq \mathcal{D}_\theta$. It is therefore sufficient to solve for values of θ such that this inclusion holds.

Suppose $z = a + ib$ where $a < 0$. Then $z \in \mathcal{D}_\theta$ provided

$$\big|1 + (a+ib)\theta\big|^2 < \big|1 - (a+ib)(1-\theta)\big|^2$$

or equivalently when

$$(1 + a\theta)^2 + (b\theta)^2 < \big(1 - a(1-\theta)\big)^2 + \big(b(1-\theta)\big)^2.$$

To solve this inequality in θ expand it to obtain

$$2a - a^2 - b^2 + 2a^2\theta + 2b^2\theta < 0 \qquad \text{or} \qquad \theta < \frac{a^2+b^2-2a}{2(a^2+b^2)} = \frac{1}{2} - \frac{a}{a^2+b^2}.$$

Since $a < 0$ then any $\theta \in [0, 1/2]$ satisfies the inequality. On the other hand, if $\theta > 1/2$ there are values of $a < 0$ for which the inequality is not satisfied. Therefore, all values of θ such that $\mathbf{C}^- \subseteq \mathcal{D}_\theta$ are given by $\theta \in [0, 1/2]$.

6. [Iserles 4.5] Prove for every ν-stage explicit Runge–Kutta method of order ν that

$$r(z) = \sum_{k=0}^{\nu} \frac{1}{k!} z^k \qquad \text{for} \qquad z \in \mathbf{C}.$$

After taking $z = h\lambda$ we have $r(z)$ is the quantity in brackets from Iserles problem 3.5:

[Iserles 3.5] Suppose that a ν-stage ERK method of order ν is applied to the linear scalar equation $y' = \lambda y$ Prove that

$$y_n = \left[\sum_{k=0}^{\nu} \frac{1}{k!}(h\lambda)^k\right]^n y_0 \qquad \text{for} \qquad n = 0, 1, \ldots.$$

First note that

$$\begin{aligned} k_1 &= f(t_n, y_n) = \lambda y_n \\ k_2 &= f(t_n + c_2 h, y_n + h a_{21} k_1) = \lambda(y_n + h a_{21}\lambda y_n) = \lambda(1 + a_{21} h\lambda) y_n \end{aligned}$$

Claim, in general that

$$k_i = \lambda p_i(h\lambda) y_n$$

where p_i is a polynomial of degree $i-1$. As this equality clearly holds for k_1 and k_2, by induction it is sufficient to show k_{i+1} follows from the equalities for $k_1, k_2, \ldots, k_i$.

$$\begin{aligned} k_{i+1} &= f\Big(t_n + c_{i+1} h, y_n + h \sum_{j=1}^{i} a_{i+1,j} k_j\Big) \\ &= \lambda\Big(y_n + h\sum_{j=1}^{i} a_{i+1,j}\lambda p_j(h\lambda) y_n\Big) \\ &= \lambda\Big(1 + \sum_{j=1}^{i} a_{i+1,j} h\lambda p_j(h\lambda)\Big) y_n = \lambda p_{i+1}(h\lambda) y_n, \end{aligned}$$

where p_{i+1} is a polynomial of degree i. This completes the induction.

The ERK method may now be expressed as

$$\begin{aligned} y_{n+1} &= y_n + h(b_1 k_1 + \cdots + b_\nu k_\nu) \\ &= y_n + h\big(b_1 \lambda p_1(h\lambda) y_n + \cdots + b_\nu \lambda p_\nu(h\lambda) y_n\big) \\ &= \big(1 + h\lambda(b_1 p_1(h\lambda) + \cdots + b_\nu p_\nu(h\lambda))\big) y_n = r(h\lambda) y_n \end{aligned}$$

where r is a polynomial of degree ν.

Now, since the method is of order ν then the truncation error $\tau_n = \mathcal{O}(h^{\nu+1})$. Plugging in the exact solution $y(t) = y_0 e^{\lambda t}$ into the the ERK method yields

$$\tau_n = y(t_{n+1}) - r(h\lambda) y(t_n) = y_0\big(e^{h\lambda} - r(h\lambda)\big)e^{\lambda t_n} = \mathcal{O}(h^{\nu+1}).$$

Solving for $r(h\lambda)$ yields that

$$r(h\lambda) = e^{h\lambda} + \mathcal{O}(h^{\nu+1}) = \sum_{k=0}^{\nu} \frac{1}{k!}(h\lambda)^k + \mathcal{O}(h^{\nu+1}).$$

Finally, since r is a polynomial of degree ν it follows exactly that

$$r(h\lambda) = \sum_{k=0}^{\nu} \frac{1}{k!}(h\lambda)^k \quad \text{and consequently} \quad y_n = r(\lambda h)^n y_0 = \left[\sum_{j=0}^{\nu} \frac{1}{k!}(h\lambda)^k\right]^n y_0.$$

7. [Iserles 4.6] Evaluate explicitly the function r for the following Runge–Kutta methods:

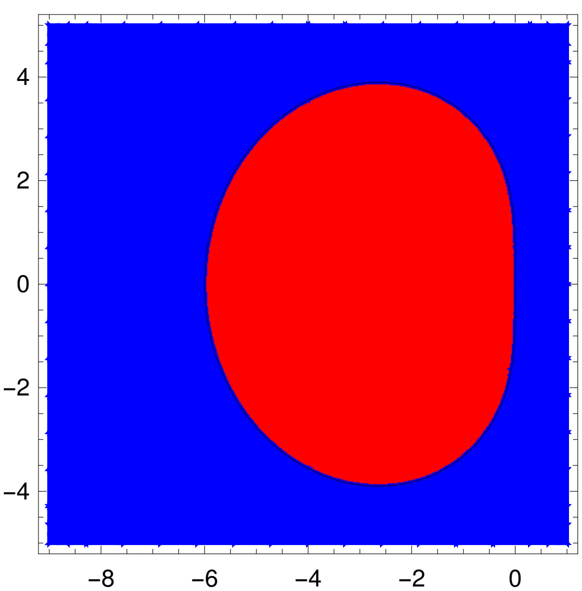
$$
\textbf{a}\quad \begin{array}{c|cc} 0 & 0 & 0 \\ \frac{2}{3} & \frac{1}{3} & \frac{1}{3} \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array}, \qquad
\textbf{b}\quad \begin{array}{c|cc} \frac{1}{6} & \frac{1}{6} & 0 \\ \frac{5}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}, \qquad
\textbf{c}\quad \begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\ 1 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}.
$$

Are any of these methods A-stable?

For method a, the script

```
eq1=k1==f[t,y[t]]
eq2=k2==f[t+2/3*h,y[t]+h*(1/3*k1+1/3*k2)]
method=y[t+h]==y[t]+h*(1/4*k1+3/4*k2)
f=Function[{t,y},lambda*y]
eq1m=eq1/.{h->1,lambda->z}
eq2m=eq2/.{h->1,lambda->z}
methodm=method/.{h->1,lambda->z}
y=Function[t,w^t]
eq1w=eq1m/.t->0
eq2w=eq2m/.t->0
methodw=methodm/.t->0
s1=Solve[{eq1w,eq2w,methodw},{k1,k2,w}]
wabs=(Abs[w]/.s1)[[1]]
w1abs=wabs/.z->a+I*b
p1=ContourPlot[w1abs,{a,-9,1},{b,-5,5},
Contours->{1},ContourShading->{Red,Blue}]
Export["hw2p7a.eps",p1,ImageSize->200]
```

produced the graph



and the output

```
Wolfram Language 12.1.1 Engine for Linux ARM (32-bit)
Copyright 1988-2020 Wolfram Research, Inc.

In[1]:= eq1=k1==f[t,y[t]]

Out[1]= k1 == f[t, y[t]]

In[2]:= eq2=k2==f[t+2/3*h,y[t]+h*(1/3*k1+1/3*k2)]

                2 h          k1   k2
Out[2]= k2 == f[--- + t, h (-- + --) + y[t]]
                 3           3    3

In[3]:= method=y[t+h]==y[t]+h*(1/4*k1+3/4*k2)

                       k1   3 k2
Out[3]= y[h + t] == h (-- + ----) + y[t]
                       4     4

In[4]:= f=Function[{t,y},lambda*y]

Out[4]= Function[{t, y}, lambda y]

In[5]:= eq1m=eq1/.{h->1,lambda->z}

Out[5]= k1 == z y[t]

In[6]:= eq2m=eq2/.{h->1,lambda->z}

                 k1   k2
Out[6]= k2 == z (-- + -- + y[t])
                 3    3

In[7]:= methodm=method/.{h->1,lambda->z}

                    k1   3 k2
Out[7]= y[1 + t] == -- + ---- + y[t]
                    4     4

In[8]:= y=Function[t,w^t]

                     t
Out[8]= Function[t, w ]
```

```
In[9]:= eq1w=eq1m/.t->0

Out[9]= k1 == z

In[10]:= eq2w=eq2m/.t->0

                    k1   k2
Out[10]= k2 == (1 + -- + --) z
                    3    3

In[11]:= methodw=methodm/.t->0

                  k1   3 k2
Out[11]= w == 1 + -- + ----
                  4     4

In[12]:= s1=Solve[{eq1w,eq2w,methodw},{k1,k2,w}]

                                 2                     2
                          3 z + z         -(6 + 4 z + z )
Out[12]= {{k1 -> z, k2 -> -(--------), w -> ---------------}}
                           -3 + z           2 (-3 + z)

In[13]:= wabs=(Abs[w]/.s1)[[1]]

                     2
             6 + 4 z + z
         Abs[------------]
                -3 + z
Out[13]= -----------------
                 2

In[14]:= w1abs=wabs/.z->a+I*b

                                        2
             6 + 4 (a + I b) + (a + I b)
         Abs[----------------------------]
                      -3 + a + I b
Out[14]= ---------------------------------
                         2

In[15]:= p1=ContourPlot[w1abs,{a,-9,1},{b,-5,5},
         Contours->{1},ContourShading->{Red,Blue}]
```

```
Out[15]= -Graphics-

In[16]:= Export["hw2p7a.eps",p1,ImageSize->200]

Out[16]= hw2p7a.eps

In[17]:=
```

From this output we deduce that

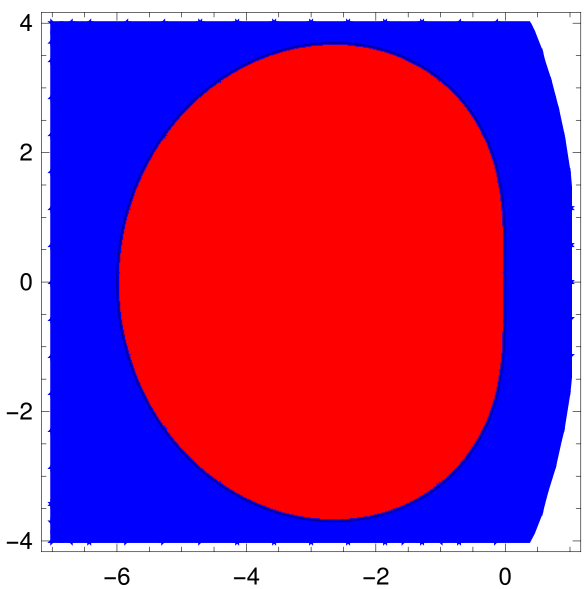
$$r(z) = -\frac{z^2 + 4z + 6}{2(z-3)}$$

but that method a is not A-stable.

For method b, we change the first three lines of the above script to read as

```
eq1=k1==f[t+1/6*h,y[t]+h*(1/6*k1)]
eq2=k2==f[t+5/6*h,y[t]+h*(2/3*k1+1/6*k2)]
method=y[t+h]==y[t]+h*(1/2*k1+1/2*k2)
```

The resulting script produced the graph



and the output

```
Wolfram Language 12.1.1 Engine for Linux ARM (32-bit)
Copyright 1988-2020 Wolfram Research, Inc.

In[1]:= eq1=k1==f[t+1/6*h,y[t]+h*(1/6*k1)]

                h       h k1
```

```
Out[1]= k1 == f[- + t, ---- + y[t]]
                6       6

In[2]:= eq2=k2==f[t+5/6*h,y[t]+h*(2/3*k1+1/6*k2)]

                5 h         2 k1   k2
Out[2]= k2 == f[--- + t, h (---- + --) + y[t]]
                 6           3     6

In[3]:= method=y[t+h]==y[t]+h*(1/2*k1+1/2*k2)

                       k1   k2
Out[3]= y[h + t] == h (-- + --) + y[t]
                       2    2

In[4]:= f=Function[{t,y},lambda*y]

Out[4]= Function[{t, y}, lambda y]

In[5]:= eq1m=eq1/.{h->1,lambda->z}

                 k1
Out[5]= k1 == z (-- + y[t])
                 6

In[6]:= eq2m=eq2/.{h->1,lambda->z}

                 2 k1   k2
Out[6]= k2 == z (---- + -- + y[t])
                  3     6

In[7]:= methodm=method/.{h->1,lambda->z}

                    k1   k2
Out[7]= y[1 + t] == -- + -- + y[t]
                    2    2

In[8]:= y=Function[t,w^t]

                     t
Out[8]= Function[t, w ]

In[9]:= eq1w=eq1m/.t->0
```

```
                   k1
Out[9]= k1 == (1 + --) z
                   6

In[10]:= eq2w=eq2m/.t->0

                    2 k1   k2
Out[10]= k2 == (1 + ---- + --) z
                     3     6

In[11]:= methodw=methodm/.t->0

                  k1   k2
Out[11]= w == 1 + -- + --
                  2    2

In[12]:= s1=Solve[{eq1w,eq2w,methodw},{k1,k2,w}]

                                          2                           2
                  -6 z         18 (2 z + z )         -36 - 24 z - 7 z
Out[12]= {{k1 -> ------, k2 -> -------------, w -> -(-----------------)}}
                 -6 + z                  2                        2
                               (-6 + z)                 (-6 + z)

In[13]:= wabs=(Abs[w]/.s1)[[1]]

                             2
              -36 - 24 z - 7 z
Out[13]= Abs[-----------------]
                         2
                 (-6 + z)

In[14]:= w1abs=wabs/.z->a+I*b

                                             2
              -36 - 24 (a + I b) - 7 (a + I b)
Out[14]= Abs[---------------------------------]
                                     2
                   (-6 + a + I b)

In[15]:= p1=ContourPlot[w1abs,{a,-7,1},{b,-4,4},
         Contours->{1},ContourShading->{Red,Blue}]

Out[15]= -Graphics-
```

```
In[16]:= Export["hw2p7b.eps",p1,ImageSize->200]

Out[16]= hw2p7b.eps

In[17]:=
```

From this output we deduce that

$$r(z) = -\frac{7z^2 + 24z + 36}{(z-6)^2}$$

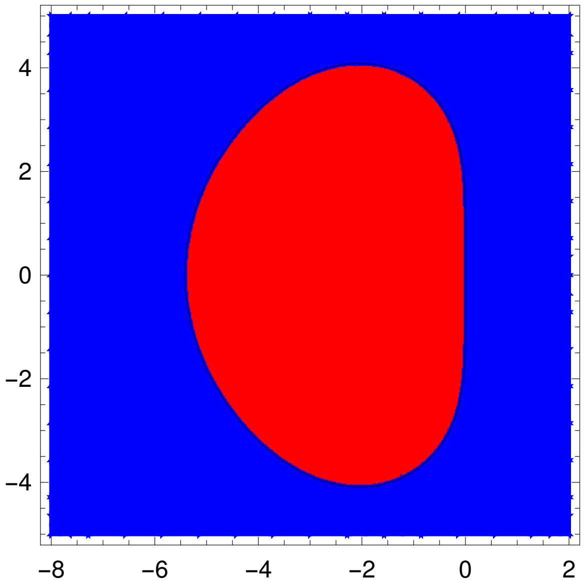
but that method b is not A-stable.

For method c, the script

```
eq1=k1==f[t,y[t]]
eq2=k2==f[t+1/2*h,y[t]+h*(1/4*k1+1/4*k2)]
eq3=k3==f[t+h,y[t]+h*k2]
method=y[t+h]==y[t]+h*(1/6*k1+2/3*k2+1/6*k3)
f=Function[{t,y},lambda*y]
eq1m=eq1/.{h->1,lambda->z}
eq2m=eq2/.{h->1,lambda->z}
eq3m=eq3/.{h->1,lambda->z}
methodm=method/.{h->1,lambda->z}
y=Function[t,w^t]
eq1w=eq1m/.t->0
eq2w=eq2m/.t->0
eq3w=eq3m/.t->0
methodw=methodm/.t->0
s1=Solve[{eq1w,eq2w,eq3w,methodw},{k1,k2,k3,w}]
wabs=(Abs[w]/.s1)[[1]]
w1abs=wabs/.z->a+I*b
p1=ContourPlot[w1abs,{a,-8,2},{b,-5,5},
Contours->{1},ContourShading->{Red,Blue}]
Export["hw2p7c.eps",p1,ImageSize->200]
```

produced the graph

Math 467/667: Homework 1 Solutions



and the output

```
Wolfram Language 12.1.1 Engine for Linux ARM (32-bit)
Copyright 1988-2020 Wolfram Research, Inc.

In[1]:= eq1=k1==f[t,y[t]]

Out[1]= k1 == f[t, y[t]]

In[2]:= eq2=k2==f[t+1/2*h,y[t]+h*(1/4*k1+1/4*k2)]

                 h          k1   k2
Out[2]= k2 == f[- + t, h (-- + --) + y[t]]
                 2          4    4

In[3]:= eq3=k3==f[t+h,y[t]+h*k2]

Out[3]= k3 == f[h + t, h k2 + y[t]]

In[4]:= method=y[t+h]==y[t]+h*(1/6*k1+2/3*k2+1/6*k3)

                       k1   2 k2   k3
Out[4]= y[h + t] == h (-- + ---- + --) + y[t]
                       6     3     6

In[5]:= f=Function[{t,y},lambda*y]

Out[5]= Function[{t, y}, lambda y]
```

```
In[6]:= eq1m=eq1/.{h->1,lambda->z}

Out[6]= k1 == z y[t]

In[7]:= eq2m=eq2/.{h->1,lambda->z}

                 k1   k2
Out[7]= k2 == z (-- + -- + y[t])
                 4    4

In[8]:= eq3m=eq3/.{h->1,lambda->z}

Out[8]= k3 == z (k2 + y[t])

In[9]:= methodm=method/.{h->1,lambda->z}

                    k1   2 k2   k3
Out[9]= y[1 + t] == -- + ---- + -- + y[t]
                    6     3     6

In[10]:= y=Function[t,w^t]

                       t
Out[10]= Function[t, w ]

In[11]:= eq1w=eq1m/.t->0

Out[11]= k1 == z

In[12]:= eq2w=eq2m/.t->0

                    k1   k2
Out[12]= k2 == (1 + -- + --) z
                    4    4

In[13]:= eq3w=eq3m/.t->0

Out[13]= k3 == (1 + k2) z

In[14]:= methodw=methodm/.t->0

                  k1   2 k2   k3
Out[14]= w == 1 + -- + ---- + --
                  6     3     6
```

```
In[15]:= s1=Solve[{eq1w,eq2w,eq3w,methodw},{k1,k2,k3,w}]

                                     2                     2    3
                              4 z + z             4 z + 3 z  + z
Out[15]= {{k1 -> z, k2 -> -(--------), k3 -> -(---------------),
                               -4 + z                 -4 + z

                              2    3
          -(24 + 18 z + 6 z  + z )
>     w -> -----------------------}}
                 6 (-4 + z)

In[16]:= wabs=(Abs[w]/.s1)[[1]]

                              2    3
            24 + 18 z + 6 z  + z
         Abs[--------------------]
                    -4 + z
Out[16]= -------------------------
                     6

In[17]:= w1abs=wabs/.z->a+I*b

                                              2            3
            24 + 18 (a + I b) + 6 (a + I b)  + (a + I b)
         Abs[-------------------------------------------]
                              -4 + a + I b
Out[17]= ------------------------------------------------
                                 6

In[18]:= p1=ContourPlot[w1abs,{a,-8,2},{b,-5,5},
         Contours->{1},ContourShading->{Red,Blue}]

Out[18]= -Graphics-

In[19]:= Export["hw2p7c.eps",p1,ImageSize->200]

Out[19]= hw2p7c.eps

In[20]:=
```

From this output we deduce that

$$r(z) = -\frac{z^3 + 6z^2 + 18z + 24}{6(z-4)}$$

but that method c is not A-stable.