Exercise 12.2. Calculate the coefficients $w_0$, $w_1$, $x_0$ and $x_1$ that make the approximation
\[ \int_0^1 xf(x) \, dx \approx w_0 f(x_0) + w_1 f(x_1) \]
exact when $f$ is any cubic polynomial. Verify your solution by letting $f$ be a general cubic polynomial. (Note that for this exercise you must find the first three orthogonal polynomials with respect to the weight function $w(x) = x$ on the interval $[0, 1]$. Then find the zeros of the quadratic for the nodes $x_0, x_1$ and use Formula (12.11) in the text to compute the weights $w_0, w_1$.)

Solution. (15) The inner product for $C[0, 1]$ that we use here is the weighted inner product
\[ \langle f, g \rangle = \int_0^1 xf(x) g(x) \, dx. \]
We first find the orthogonal polynomials with respect to this weight by setting
\begin{align*}
p_0(x) &= 1 \\
p_1(x) &= x - \frac{\langle p_0, x \rangle}{\langle p_0, p_0 \rangle} p_0(x) = x - \frac{\int_0^1 x \cdot 1 \cdot x \, dx}{\int_0^1 x \cdot 1 \cdot 1 \, dx} \\
&= x - \frac{2}{3} \\
p_2(x) &= x^2 - \frac{\langle p_0, x^2 \rangle}{\langle p_0, p_0 \rangle} p_0(x) - \frac{\langle p_1, x \rangle}{\langle p_1, p_1 \rangle} p_1(x) \\
&= x^2 - \frac{\int_0^1 x \cdot 1 \cdot x^2 \, dx}{\int_0^1 x \cdot 1 \cdot 1 \, dx} - \frac{\int_0^1 x \cdot 1 \cdot x \, dx}{\int_0^1 x \cdot 1 \cdot 1 \, dx} x^2 \\
&= x^2 - \frac{3}{10} - \frac{6}{5} x = x^2 - \frac{6}{5} x + \frac{3}{10}.
\end{align*}
The nodes are just the zeros of $p_2(x)$, which are given by
\[ x = \frac{6}{5} \pm \sqrt{\frac{36}{25} - 4 \cdot \frac{3}{10}} = \frac{6 \pm \sqrt{6}}{10}. \]
Take $x_0 = (6 - \sqrt{6})/10$ and $x_1 = (6 + \sqrt{6})/10$, so that
\[ x_1 - x_0 = \frac{\sqrt{6}}{5}. \]
The weights are then given by

\[ w_0 = \int_0^1 \frac{x - x_1}{x_0 - x_1} \, dx = \frac{1}{4} - \frac{\sqrt{6}}{36} \]

\[ w_1 = \int_0^1 \frac{x - x_0}{x_1 - x_0} \, dx = \frac{1}{4} + \frac{\sqrt{6}}{36} \]

Now we can check by exact arithmetic or simply by Matlab that

\[ w_0 + w_1 = \frac{1}{2} = \int_0^1 x \cdot 1 \, dx \]

\[ w_0 x_0 + w_1 x_1 = \frac{1}{3} = \int_0^1 x \cdot x \, dx \]

\[ w_0 x_0^2 + w_1 x_1^2 = \frac{1}{4} = \int_0^1 x \cdot x^2 \, dx \]

\[ w_0 x_0^3 + w_1 x_1^3 = \frac{1}{5} = \int_0^1 x \cdot x^3 \, dx \]

Thus the formula is exact on any linear combination of these, which amounts to any cubic polynomial.

**Exercise N5.** Calculate a Padé approximation for \( f(x) = e^x \) of total degree \( 3 = N = n + m \), with \( n = 2 \) and \( m = 1 \). (See the ClassroomNotes441.pdf file for an explanation of the notation.) Next find the best cubic approximation to \( f(x) \) on \([-1, 1]\). Find the (approximate) max norm of the error with Matlab. Which is better? (See our ClassroomNotes file for more information on Padé approximations.)

**Solution.** (15) The form of the approximation is

\[ r(x) = \frac{p(x)}{q(x)} = \frac{p_0 + p_1 x + p_2 x^2}{q_0 + q_1 x}, \]

where \( q_0 = 1 \) and all higher terms \( p_i, i > 2 \) \( q_j, j > 1 \), are zero. Furthermore, we know that

\[ e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \cdots = a_0 + a_1 x + a_2 x^2 + \cdots. \]

The equations that we need for the remaining 4 terms are given as in the ClassroomNotes file as

\[ 0 = q_0 a_0 - p_0 = 1 \cdot 1 - p_0 = 1 - p_0 \]
\[ 0 = q_0 a_1 + q_1 a_0 - p_1 = 1 + q_1 - p_1 \]
\[ 0 = q_0 a_2 + q_1 a_1 + q_2 a_0 - p_2 = \frac{1}{2} + q_1 + 0 - p_2 \]
\[ 0 = q_0 a_3 + q_1 a_2 + q_2 a_1 + q_3 a_0 - p_3 = \frac{1}{6} + \frac{1}{2}q_1 + 0 - 0 \]

which gives \( p_0 = 1, p_1 - q_1 = 1, p_2 - q_1 = \frac{1}{2}, \) and \(-\frac{1}{2}q_1 = \frac{1}{6}\). It follows that \( q_1 = -1/3, p_2 = 1/6, p_1 = 2/3 \) and \( p_0 = 1 \). Hence the Padé approximation is

\[ r(x) = \frac{1 + \frac{2}{3}x + \frac{1}{6}x^2}{1 - \frac{1}{3}x}. \]
Next, we use minimaxd.m to find the best cubic approximation and compare it to the Pade approximation. Here is the transcript:

```
octave:1> x = -1:0.001:1;
octave:2> expx = exp(x);
octave:3> bestcubic = minimaxd(expx,x,3)
bestcubic =
0.17953 0.54297 0.99567 0.99458
octave:4> cubicerr = norm(expx-polyval(bestcubic,x),inf)
cubicerr = 0.0055284
octave:5> pade = (1+2*x/3+x.^2/6)./(1-x/3);
octave:6> padeerr = norm(expx-pade,inf)
padeerr = 0.031718
```

(An examination of the graphs shows the problem: The Pade approximation does much better on the interval $[-0.5, 0.5]$, but rapidly loses accuracy at points outside this interval about 0, just like a Taylor approximation. On the other hand, if you were to plot the Taylor cubic polynomial, you see that the Pade approximation does much better than the Taylor polynomial.)

**Exercise N6.** Determine by Matlab experiment how many function evaluations are required to approximate $\int_{-1}^{1} e^x \sin(\pi x) dx$ to an accuracy as good as one obtains with a 7 point Gaussian quadrature method, if one uses a trapezoidal method with equally spaced nodes. (Exact answer: $\left(\frac{e - 1/e}{\pi + 1/\pi}\right)$.)

**Solution.** (10) We recode ‘myfcn.m’ so that the integrand becomes myfcn:

```
retval = exp(x).*sin(pi*x)
```

Then we use the function file GaussInt.m to calculate the 7 point quadrature approximation to the integral. Experimentation with the function file trap.m eventually yields that an 9182 point trapezoidal method just exceeds the accuracy of the Gaussian quadrature result.

```
octave:1> format long
octave:2> gauss7 = GaussInt(@myfcn,[-1,1],7)
gauss7 = 0.679326212604478
octave:3> truevalue = (exp(1)-1/exp(1))/(pi+1/pi)
truevalue = 0.679326183402095
octave:4> abs(gauss7-truevalue)
ans = 2.920238284520e-08
octave:5> n = 9181; x = linspace(-1,1,n); abs(trap(x,myfcn(x))-truevalue)
ans = 2.92069020080277e-08
octave:6> n = 9182; x = linspace(-1,1,n); abs(trap(x,myfcn(x))-truevalue)
ans = 2.92005395419181e-08
```