EX.3.1.5, Sauer3

a. Find a polynomial $p(x)$ of degree 3 or less whose graph passes through the four data points $(-2, 8)$, $(0, 4)$, $(1, 2)$, $(3, -2)$.

b. Describe any other polynomials of degree 4 or less which pass through the four points in part (a).

EX.3.1.5, Sauer3, solution, Langou

For Part (a), the problem does not specify which method to use. So we can use any method: (1) Lagrange, (2) "fake" divided difference, (3) divided difference, or (4) "Vandermonde". It turns out the four points are on a line. So that the polynomial of degree at least 3 that interpolates the four points is a line. Using Vandermonde is rarely a good idea, we do it anyway. In this case, Lagrange method is not great. It turns out that divided difference gives a quick and easy answer. See at the end for all four methods used on Part (a).

Colab: `https://colab.research.google.com/drive/1PdkL7_0yW7CCBqj22YlVIGc2GwQDbsWh`

Part (a)

The data points are $(-2, 8)$, $(0, 4)$, $(1, 2)$, $(3, -2)$. We use Newton's divided difference. We get

$$\begin{array}{r|rrrr} -2 & 8 & & & \\ 0 & 4 & -2 & & \\ 1 & 2 & -2 & 0 & \\ 3 & -2 & -2 & 0 & 0 \end{array}$$

Finally:

$$\boxed{p_3(x) = 8 - 2(x + 2) = 4 - 2x.}$$

Part (b)

All polynomials of degree 4 or less which pass through the four points $(-2, 8)$, $(0, 4)$, $(1, 2)$, $(3, -2)$ are of the form

$$\boxed{p_4(x) = 4 - 2x + \alpha(x + 2)x(x - 1)(x - 3), \qquad \text{for any value of } \alpha.}$$

What is below is not needed

It should be clear that

a. $p_4(x)$ is of degree 4 or less,

b. $p_4(x)$ interpolates the four data points $(-2, 8)$, $(0, 4)$, $(1, 2)$, and $(3, -2)$.

It might not be so clear that

c. $p_4(x)$ represents all polynomials of degree 4 or less which pass through the four points $(-2, 8)$, $(0, 4)$, $(1, 2)$, $(3, -2)$.

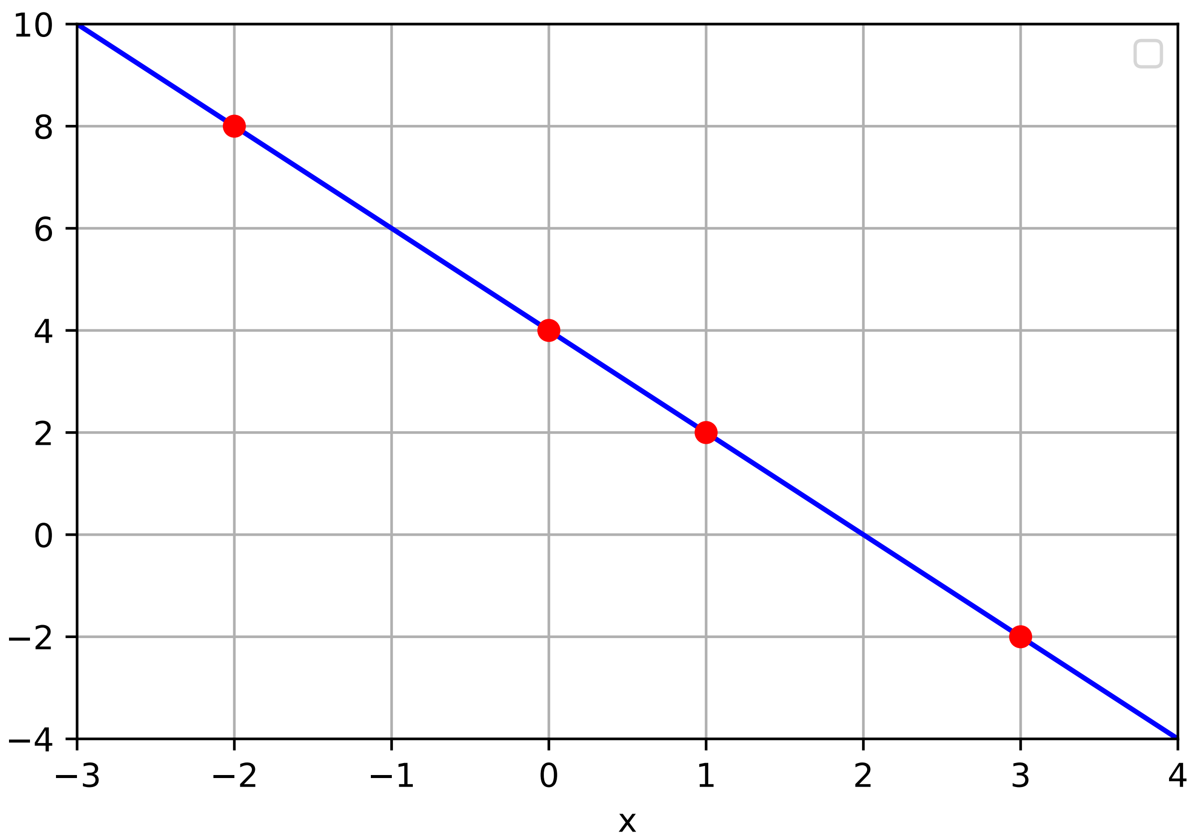
The reasoning is explained in Sauer.

What is below is not needed

```
x = np.array([ -2., 0., 1.,  3.])
y = np.array([  8., 4., 2., -2.])

print( "data points to interpolate:")
print( "    |    x       |    y       |" )
for i in range(0,len(x)):
    print( f"{i:2d}", " | ", f"{x[i]:6.3f}", " | ", f"{y[i]:6.3f}", " | " )

# use np.polynomial.polynomial.polyfit to get the interpolating polynomial
# remember that `numpy` polynomial coefficients are stored from degree 0 to the
c = newtdd(x,y)
print( "coefficient of p(x):    ")
for i in range(0,len(c)):
    print( "c[",i,"] = ", f"{c[i]: 20.16f}")

# check that the interpolating polynomial correctly interpolates our data points
# we evaluate p at x and we check that we get (exactly) y back
yy = polyval_nested_w_base_points(c,x,x)
err = abs( yy - y )
print("------------------------------------------------------------- -----")
print("|    x     |          y          |          p(x)        |  error
|")
print("-------------------------------------------------------------------")
for i in range(0,len(x)):
    print("|",f"{x[i]: 7.5f}","|", f"{y[i]: 20.16f}","|", f"{yy[i]: 20.16f}","|"
print("-------------------------------------------------------------------")


# plot
xx = np.linspace( -3, 4, 1000)
yy = polyval_nested_w_base_points(c,x,xx)
plt.plot(xx, yy, '-b' )
yy = polyval_nested_w_base_points(c,x,x)
plt.plot( x, yy, 'ro')
plt.xlabel('x')
plt.xlim([ -3.,  4. ])
plt.ylim([ -4., 10. ])
plt.grid()
plt.show()
```

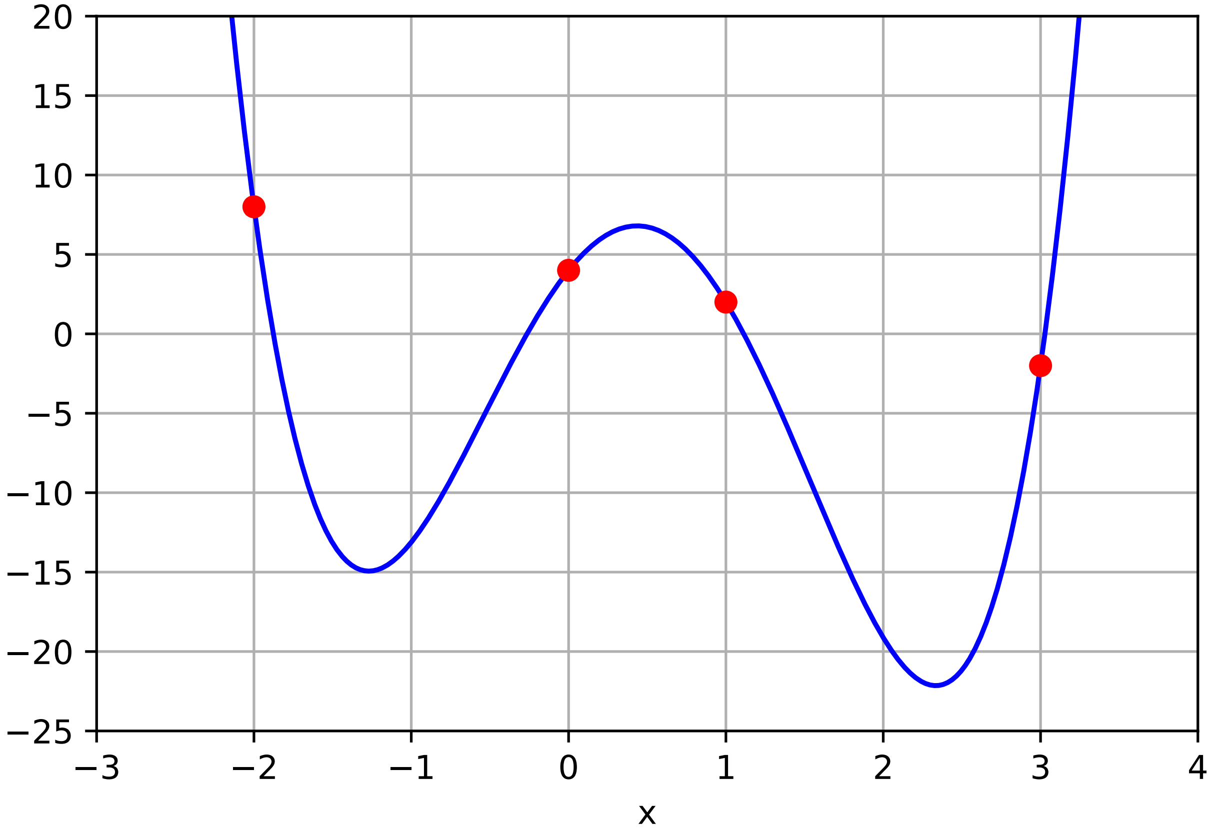
```
data points to interpolate:
```

```
    |     x      |    y      |
 0  |   -2.000   |   8.000   |
 1  |    0.000   |   4.000   |
 2  |    1.000   |   2.000   |
 3  |    3.000   |  -2.000   |
coefficient of p(x):
c[ 0 ] =    8.000000000000000
c[ 1 ] =   -2.000000000000000
c[ 2 ] =    0.000000000000000
c[ 3 ] =    0.000000000000000
------------------------------------------------------------------------ -----
|     x     |            y            |           p(x)           |  error   |
-------------------------------------------------------------------------------
| -2.00000  |    8.000000000000000    |    8.000000000000000     |  0.0e+00 |
|  0.00000  |    4.000000000000000    |    4.000000000000000     |  0.0e+00 |
|  1.00000  |    2.000000000000000    |    2.000000000000000     |  0.0e+00 |
|  3.00000  |   -2.000000000000000    |   -2.000000000000000     |  0.0e+00 |
-------------------------------------------------------------------------------
```




```
# we take alpha = 2.39, but please change alpha as you want and, for each alpha,
# you get a different polynomial of degree 4 or less which pass through
# the four points $(-2,8)$, $(0,4)$, $(1,2)$, $(3,-2)$.

c = newtdd(x,y)

alpha = 2.39
```

```
c = np.append( c, alpha )

print( "coefficient of p(x):    ")
for i in range(0,len(c)):
    print( "c[",i,"] = ", f"{c[i]: 20.16f}")

# check that the interpolating polynomial correctly interpolates our data points
# we evaluate p at x and we check that we get (exactly) y back
yy = polyval_nested_w_base_points(c,x,x)
err = abs( yy - y )
print("---------------------------------------------------------------------- -----")
print("|    x     |           y            |           p(x)         |  error
|")
print("----------------------------------------------------------------------------")
for i in range(0,len(x)):
    print("|",f"{x[i]: 7.5f}","|", f"{y[i]: 20.16f}","|", f"{yy[i]: 20.16f}","|"
print("----------------------------------------------------------------------------")


# plot
xx = np.linspace( -3, 4, 1000)
yy = polyval_nested_w_base_points(c,x,xx)
plt.plot(xx, yy, '-b' )
yy = polyval_nested_w_base_points(c,x,x)
plt.plot( x, yy, 'ro')
plt.xlabel('x')
plt.xlim([ -3.,  4. ])
plt.ylim([ -25., 20. ])
plt.grid()
plt.show()
```



What is below is not needed

Part (a) - (1) Lagrange

For the points $(-2, 8)$, $(0, 4)$, $(1, 2)$, $(3, -2)$, we have

$$\begin{aligned}
\ell_1(x) &= \frac{(x-0)(x-1)(x-3)}{(-2-0)(-2-1)(-2-3)} = -\frac{1}{30}x(x-1)(x-3);\\
\ell_2(x) &= \frac{(x+2)(x-1)(x-3)}{(0+2)(0-1)(0-3)} = \frac{1}{6}(x+2)(x-1)(x-3);\\
\ell_3(x) &= \frac{(x+2)(x-0)(x-3)}{(1+2)(1-0)(1-3)} = -\frac{1}{6}(x+2)x(x-3);\\
\ell_4(x) &= \frac{(x+2)(x-0)(x-1)}{(3+2)(3-0)(3-1)} = \frac{1}{30}(x+2)x(x-1).
\end{aligned}$$

This leads to

$$\begin{aligned}
p(x) &= 8\cdot\ell_1(x) + 4\cdot\ell_2(x) + 2\cdot\ell_3(x) - 2\cdot\ell_3(x)\\
&= -\frac{8}{30}x(x-1)(x-3) + \frac{4}{6}(x+2)(x-1)(x-3) - \frac{2}{6}(x+2)x(x-3) - \frac{2}{30}(x+2)x(x-1)
\end{aligned}$$

$$\boxed{p_3(x) = -\tfrac{8}{30}x(x-1)(x-3) + \tfrac{4}{6}(x+2)(x-1)(x-3) - \tfrac{2}{6}(x+2)x(x-3) - \tfrac{2}{30}(x+2)x(x-1).}$$

We can also work on developing this to get the “natural” form

$$\begin{aligned}
p(x) &= -\frac{8}{30}x(x-1)(x-3) + \frac{4}{6}(x+2)(x-1)(x-3) - \frac{2}{6}(x+2)x(x-3) - \frac{2}{30}(x+2)x(x-1) \\
&= -\frac{8}{30}(x^3-4x^2+3x) + +\frac{4}{6}(x^3-2x^2-5x+6) - \frac{2}{6}(x^3-x^2-6x) - \frac{2}{30}(x^3+x^2-2x) \\
&= (-\frac{8}{30}+\frac{4}{6}-\frac{2}{6}-\frac{2}{30})x^3 + (\frac{32}{30}-\frac{8}{6}+\frac{2}{6}-\frac{2}{30})x^2 + (-\frac{24}{30}-\frac{20}{6}+\frac{12}{6}+\frac{4}{30})x + \frac{24}{6} \\
&= -2x+4
\end{aligned}$$

$$\boxed{p_3(x) = 4-2x.}$$

Part (a) - (2) "fake" divided difference

The data points are $(-2,8)$, $(0,4)$, $(1,2)$, $(3,-2)$.
We seek the polynomial of degree 0, $p_0(x)$, that interpolates the point $(-2,8)$. We have

$$p_0(x) = c_0$$

And so $p_0(-2) = 8$ leads to

$$c_0 = 8.$$

And we get

$$p_0(x) = 8.$$

We seek the polynomial of degree 1, $p_1(x)$, that interpolates the points $(-2,8)$ and $(0,4)$. We take p_1 of the form

$$p_1(x) = p_0(x) + c_1(x+2).$$

That way, since $p_0(-2) = 8$, we have $p_1(-2) = 8$. Since we also want $p_1(0) = 4$, we get

$$p_1(0) = 4 \quad \Rightarrow \quad 8 + c_1(0+2) = 4 \quad \Rightarrow \quad c_1 = -2.$$

And we get

$$p_1(x) = 8 - 2(x+2).$$

We seek the polynomial of degree 2, $p_2(x)$, that interpolates the points $(-2,8)$, $(0,4)$, and $(1,2)$. We take p_2 of the form

$$p_2(x) = p_1(x) + c_2(x+2)x.$$

That way, since $p_1(-2) = 8$ and $p_1(0) = 4$, we have $p_2(-2) = 8$ and $p_2(0) = 4$. Since we also want $p_2(1) = 2$, we get

$$p_2(1) = 2 \quad \Rightarrow \quad 8 - 2(1+2) + c_2(1+2)(1) = 2 \quad \Rightarrow \quad c_2 = 0.$$

And we get

$$p_2(x) = 8 - 2(x+2).$$

We seek the polynomial of degree 3, $p_3(x)$, that interpolates the points $(-2,8)$, $(0,4)$, $(1,2)$, and $(3,-2)$. We take p_3 of the form

$$p_3(x) = p_2(x) + c_3(x+2)x(x-1).$$

That way, since $p_2(-2) = 8$, $p_2(0) = 4$, and $p_2(1) = 2$, we have $p_3(-2) = 8$, $p_3(0) = 4$, and $p_3(1) = 2$. Since we also want $p_2(3) = -2$, we get

$$p_3(3) = -2 \quad \Rightarrow \quad 8 - 2(3+2) + c_3(3+2)(3)(3-1) = -2 \quad \Rightarrow \quad c_3 = 0.$$

And we get

$$p_3(x) = 8 - 2(x+2).$$

$$\boxed{p_3(x) = 8 - 2(x+2) = 4 - 2x.}$$

Part (a) - (3) divided difference

The data points are $(-2, 8)$, $(0, 4)$, $(1, 2)$, $(3, -2)$.
We use Newton's divided difference.
First step, we set up the table with $x_0 = -2$, $x_1 = 0$, $x_2 = 1$, $x_3 = 3$ and $f[x_0] = y_0 = 8$, $f[x_1] = y_1 = 4$, $f[x_2] = y_2 = 2$, $f[x_3] = y_3 = -2$. We get

$$\begin{array}{r|r} -2 & 8 \\ 0 & 4 \\ 1 & 2 \\ 3 & -2 \end{array}$$

Second step, we compute

$$\begin{aligned} f[x_0, x_1] &= \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{4-8}{0+2} = -2 \\ f[x_1, x_2] &= \frac{f[x_2] - f[x_1]}{x_2 - x_1} = \frac{2-4}{1-0} = -2 \\ f[x_2, x_3] &= \frac{f[x_3] - f[x_2]}{x_3 - x_2} = \frac{-2-2}{3-1} = -2 \end{aligned}$$

We get

$$\begin{array}{r|rr} -2 & 8 & \\ 0 & 4 & -2 \\ 1 & 2 & -2 \\ 3 & -2 & -2 \end{array}$$

For the third step, we compute

$$\begin{aligned} f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{-2+2}{1+2} = 0 \\ f[x_1, x_2, x_3] &= \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} = \frac{-2+2}{3-0} = 0 \end{aligned}$$

We get

$$\begin{array}{r|rrr} -2 & 8 & & \\ 0 & 4 & -2 & \\ 1 & 2 & -2 & 0 \\ 3 & -2 & -2 & 0 \end{array}$$

For the fourth step, we compute

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} = \frac{0-0}{3+2} = 0$$

We get

$$\begin{array}{r|rrrr} -2 & 8 & & & \\ 0 & 4 & -2 & & \\ 1 & 2 & -2 & 0 & \\ 3 & -2 & -2 & 0 & 0 \end{array}$$

We get the coefficients c_0, c_1, c_2, and c_3 by reading the diagonal of the table:

$$\begin{array}{r|rrrr} -2 & 8 & & & \\ 0 & 4 & -2 & & \\ 1 & 2 & -2 & 0 & \\ 3 & -2 & -2 & 0 & 0 \end{array}$$

And so the nested form representation

$$p(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + c_3(x - x_0)(x - x_1)(x - x_2)$$

gives

$$p(x) = 8 - 2(x+2) + 0(x+2)(x - 0) + 0(x+2)(x - 0)(x - 1)$$

Finally:

$$\boxed{p_3(x) = 8 - 2(x + 2) = 4 - 2x.}$$

Part (a) - (4) “Vandermonde”

Since we will get a 4x4 system of equations, using “Vandermonde” is really not a good idea. We really should use one of the other methods above. But doing it anyway.

The data points are $(-2, 8)$, $(0, 4)$, $(1, 2)$, $(3, -2)$.

We want to find a_0, a_1, a_1, a_3 such that $a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3 = y_i$ for the four data points (x_i, y_i): $(-2, 8)$, $(0, 4)$, $(1, 2)$, and $(3, -2)$. This gives

$$\left\{ \begin{array}{rcrcrcrcrcr} (-2,8) & : & a_0 & - & 2a_1 & + & 4a_2 & - & 8a_3 & = & 8 \\ (0,4) & : & a_0 & & & & & & & = & 4 \\ (1,2) & : & a_0 & + & a_1 & + & a_2 & + & a_3 & = & 2 \\ (3,-2) & : & a_0 & + & 3a_1 & + & 9a_2 & + & 27a_3 & = & -2 \end{array} \right.$$

$$\left(\begin{array}{rrrr|r} 1 & -2 & 4 & -8 & 8 \\ 1 & 0 & 0 & 0 & 4 \\ 1 & 1 & 1 & 1 & 2 \\ 1 & 3 & 9 & 27 & -2 \end{array} \right) \rightsquigarrow \left(\begin{array}{rrrr|r} 1 & 0 & 0 & 0 & 4 \\ 0 & -2 & 4 & -8 & 4 \\ 0 & 1 & 1 & 1 & -2 \\ 0 & 3 & 9 & 27 & -6 \end{array} \right) \rightsquigarrow \left(\begin{array}{rrrr|r} 1 & 0 & 0 & 0 & 4 \\ 0 & 1 & -2 & 4 & -2 \\ 0 & 0 & 3 & -3 & 0 \\ 0 & 0 & 3 & 15 & 0 \end{array} \right) \rightsquigarrow \left(\begin{array}{rrrr|r} 1 & 0 & 0 & 0 & 4 \\ 0 & 1 & 0 & 0 & -2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{array} \right)$$

$$\rightsquigarrow \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 4 \\ -2 \\ 0 \\ 0 \end{pmatrix}$$

$$\boxed{p_3(x) = 4 - 2x.}$$