1. The Continuation method and Eulers method gives:
   (a) $(3, -2.25)^t$
   (b) $(0.42105263, 2.6184211)^t$
   (c) $(2.173110, -1.3627731)^t$

2. The Continuation method and Eulers method gives:
   (a) $(2.3039880, -2.0010995)^t$
   (b) $(0.59709702, 2.2579684)^t$
   (c) $(2.1094460, -1.3345633)^t$

3. Using the Continuation method and Eulers method gives:
   (a) $(0.44006047, 1.8279835)^t$
   (b) $(-0.41342613, 0.096669468)^t$
   (c) $(0.49858909, 0.24999091, -0.52067978)^t$
   (d) $(6.1935484, 18.532258, -21.725806)^t$

4. (a) $(-15.78432724, 5.29974589)^t$ is not comparable using $\mathrm{x}(0) = (0, 0)^t$ as starting value. Using the starting values as in 10.2 Exercise 5(c) gives:
   $\mathrm{x}(0) = (-1, 3.5)^t$ leads to $(-1, 3.5)^t$, and $\mathrm{x}(0) = (2.5, 4)^t$ leads to $(2.54694647, 3.9849976)^t$
   (b) $(0.12124195, 0.27110516)^t$ using $\mathrm{x}(0) = (0.11, 0.27)^t$ is comparable to Newton's method. Using $\mathrm{x}(0) = (0, 0)^t$ leads to an error in the program.
   (c) $(1.03645880, 1.08572502, 0.93136714)^t$ is comparable to Newton's method.
   (d) Using $\mathrm{x}(0) = (0, 0, 0)^t$ does not allow computation of $\mathrm{x}(1)$. Using $\mathrm{x}(0) = (1, -1, 1)^t$ gives $(0.90016074, -1.00238008, 0.49661093)^t$ which is nearly comparable to Newton's method.

5. The Continuation method and the RungeKutta method of order four gives:
   (a) With $\mathrm{x}(0) = (-1, 3.5)^t$ the result is $(-1, 3.5)^t$.
   With $\mathrm{x}(0) = (2.5, 4)^t$ the result is $(2.54694647, 3.98499746)^t$.
   (b) With $\mathrm{x}(0) = (0.11, 0.27)^t$ the result is $(0.12124195, 0.27110516)^t$.
   (c) With $\mathrm{x}(0) = (1, 1, 1)^t$ the result is $(1.03640047, 1.08570655, 0.93119144)^t$.
   (d) With $\mathrm{x}(0) = (1, -1, 1)^t$ the result is $(0.90016074, -1.00238008, 0.496610937)^t$.
   With $\mathrm{x}(0) = (1, 1, -1)^t$ the result is $(0.50104035, 1.00238008, -0.49661093)^t$.

6. The Continuation method and the RungeKutta method of order four gives:

   (a) $(0.49950451, 0.86635691)^t$. This result is comparable since it required only 4 matrix inversions to obtain an answer almost as accurate as in Section 10.2 Exercise 3a with 5 iterations.

   (b) $(1.7730066, 1.7703057)^t$. This result is comparable since it required only 4 matrix inversions to obtain an answer almost as accurate as in Section 10.2 Exercise 3b with 6 iterations.

   (c) $(-1.4569217, -1.6645292, 0.42138616)^t$. This result is comparable to the result obtained in Section 10.2 Exercise 3c since it required only 4 matrix inversions as compared to 5 iterations of Newton's method.

   (d) $(0.49813364, -0.19957917, -0.52882773)^t$. This result is comparable to the result obtained in Section 10.2 Exercise 3d.

7. The Continuation method and the RungeKutta method of order four gives:

   (a) With $\mathbf{x}(0) = (-1, 3.5)^t$ the result is $(-1, 3.5)^t$.
   With $\mathbf{x}(0) = (2.5, 4)^t$ the result is $(2.5469465, 3.9849975)^t$.

   (b) With $\mathbf{x}(0) = (0.11, 0.27)^t$ the result is $(0.12124191, 0.27110516)^t$.

   (c) With $\mathbf{x}(0) = (1, 1, 1)^t$ the result is $(1.03640047, 1.08570655, 0.93119144)^t$.

   (d) With $\mathbf{x}(0) = (1, -1, 1)^t$ the result is $(0.90015964, -1.00021826, 0.49968944)^t$.
   With $\mathbf{x}(0) = (1, 1, -1)^t$ the result is $(0.5009653, 1.00021826, -0.49968944)^t$.

8. Using $\mathbf{x}(0) = (1, 1, 1, 1)^t$ gives

$$\mathbf{x}(1) = \left(10^{-10}, 0.7047619049, 0.7047619049, 1\right)^t.$$

   Using $\mathbf{x}(0) = (1, 0, 0, 0)^t$ gives

$$\mathbf{x}(1) = (0.8171787148, 0.4035113851, -0.4035113850, 2.993229684)^t.$$

   Using $\mathbf{x}(0) = (1, -1, 1, -1)^t$ gives

$$\mathbf{x}(1) = (0.5769841387, -0.5769841239, 0.5769841246, 6.019603162)^t.$$

   The other three solutions follow easily from Exercise 6(a) of Section 10.2.

9. The Continuation method and the RungeKutta method of order four gives the approximate solution, $(0.50024553, 0.078230039, -0.52156996)^t$

10. The system of differential equations to solve by Euler's method is

$$\mathbf{x}'(\lambda) = -[J(\mathbf{x}(\lambda))]^{-1}F(\mathbf{x}(0)).$$

With $N = 1$, we have $h = 1$ and

$$\mathbf{x}(1) = \mathbf{x}(0) + h[-J(\mathbf{x}(0))]^{-1}F(\mathbf{x}(0)) = \mathbf{x}(0) - hJ(\mathbf{x}(0))^{-1}F(\mathbf{x}(0)) = \mathbf{x}(0) - J(\mathbf{x}(0))^{-1}F(\mathbf{x}(0)).$$

However, Newton's method gives

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - J\left(\mathbf{x}^{(0)}\right)^{-1} F\left(\mathbf{x}^{(0)}\right).$$

Since $\mathbf{x}(0) = \mathbf{x}^{(0)}$, we have $\mathbf{x}(1) = \mathbf{x}^{(1)}$.

11. For each λ, we have

$$0 = G(\lambda, \mathbf{x}(\lambda)) = F(\mathbf{x}(\lambda)) - e^{-\lambda}F(\mathbf{x}(0)),$$

so

$$0 = \frac{\partial F(\mathbf{x}(\lambda))}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\lambda} + e^{-\lambda}F(\mathbf{x}(0)) = J(\mathbf{x}(\lambda))\mathbf{x}'(\lambda) + e^{-\lambda}F(\mathbf{x}(0))$$

and

$$J(\mathbf{x}(\lambda))\mathbf{x}'(\lambda) = -e^{-\lambda}F(\mathbf{x}(0)) = -F(\mathbf{x}(0)).$$

Thus

$$\mathbf{x}'(\lambda) = -J(\mathbf{x}(\lambda))^{-1}F(\mathbf{x}(0)).$$

With $N = 1$, we have $h = 1$ so that

$$\mathbf{x}(1) = \mathbf{x}(0) - J(\mathbf{x}(0))^{-1}F(\mathbf{x}(0)).$$

However, Newton's method gives

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - J(\mathbf{x}^{(0)})^{-1}F(\mathbf{x}^{(0)}).$$

Since $\mathbf{x}(0) = \mathbf{x}^{(0)}$, we have $\mathbf{x}(1) = \mathbf{x}^{(1)}$.

12. (a) The CMRK4 algorithm with $N = 1$ requires the solution of 4 linear systems, which is almost as much work as required for 4 iterations of Newton's method.
Exercises 5, 6, and 8 yield appropriate comparisons. In only 5a, 5b, and 5c was CMRK4 competitive with Newton's method. This suggests that CMRK4 with $N = 1$ is not as good as Newton's Method.

(b) Generally, the CMRK4 algorithm would yield good initial approximations for Newton's method. This is well illustrated in Exercises 4b, 4c, 4d, 5, 6, and 8.

(c) The CMRK4 algorithm with $N = 2$ requires the solution of 8 linear systems, which is almost as much work as required for 8 iterations of Newton's method. Exercises 7 and 9 yield appropriate comparisons. Newton's method outperformed CMRK4 in Exercise 7. The CMRK4 algorithm worked well in Exercise 9 which had the singular Jacobian. The results here suggest that CMRK4 with $N = 2$ is not as good as Newton's method.

(d) Since the CMRK4 algorithm with $N = 1$ generally yields good initial approximations for Newton's method, we would not need to use the CMRK4 algorithm with $N = 2$ for this purpose.