DATA 609: Homework 11

Modeling with Systems of Differential Equations

Aaron Grzasko

November 3, 2017

Page 529, Question 1

Verify that the given function pair is a solution to the first-order system.

$$x = -e^{-t},\ y = e^{t}$$

$$\frac{dx}{dt} = -y,\ \frac{dy}{dt} = -x$$

We can verify the solution is valid by taking the derivatives of both x and y with respect to t:

$$\frac{dx}{dt} = \frac{d}{dt}(-e^{-t}) = e^{t} = y$$

$$\frac{dy}{dt} = \frac{d}{dt}(e^{t}) = e^{t} = x$$

Page 529 Question 6

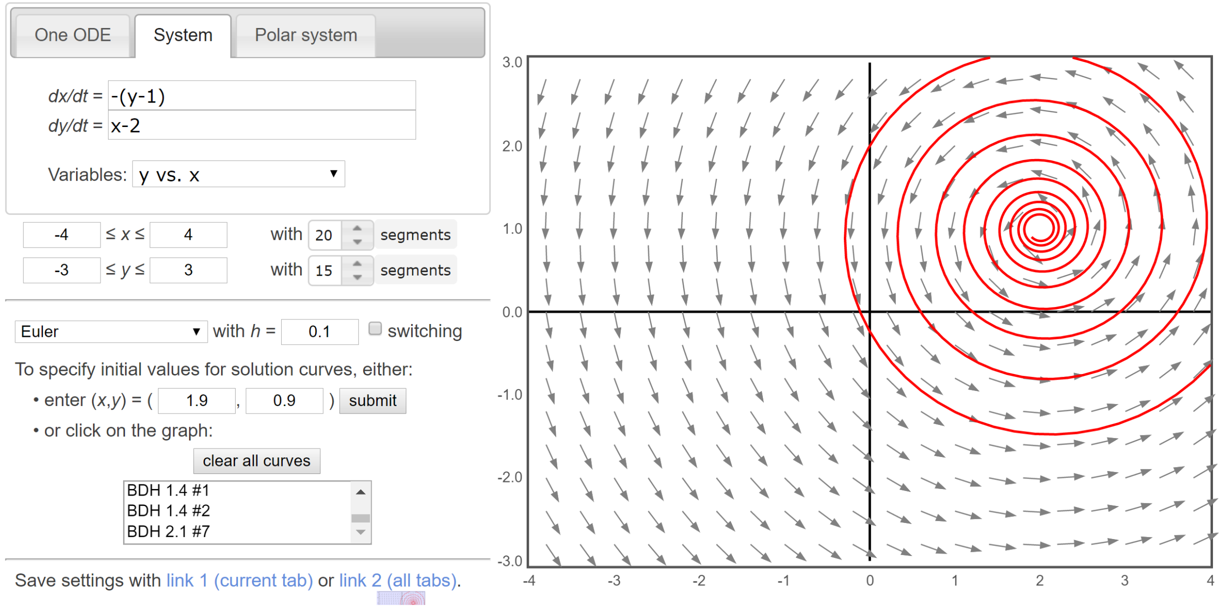
Find and classify the rest points of the given autonomous system

$$\frac{dx}{dt} = f(x,y) = -(y-1), \quad \frac{dy}{dt} = g(x,y) = x-2$$

The only rest point, i.e. where both $f(x,y) = 0$ and $g(x,y) = 0$, is at $(2,1)$.

The rest point is unstable as points close to this value form a circular, spiral trajectory with increasing radius as t increases.

Let's visualize the slope field and example trajectory near $(2,1)$, using a tool available at https://www.bluffton.edu/homepages/facstaff/nesterd/java/slopefields.html:



Page 546 Question 1

Apply the first and second derivative tests to the function $f(y) = y^a/e^{by}$ to show that $y = a/b$ is a unique critical point that yields the relative maximum $f(a/b)$.

First, find $\frac{d}{dy}(y^a/e^{by})$ using the quotient rule:

$$\frac{d}{dy}(y^a/e^{by}) = \frac{ay^{a-1}e^{by} - by^a e^{by}}{e^{2by}}$$

$$= \frac{y^{a-1}(a - by)}{e^{by}}$$

From the equation above, we determine two critical points where $\frac{d}{dy}(y^a/e^{by}) = 0$:

$y = 0$ and $y = a/b$.

Now find the second derivative:

$$\frac{d}{dy}(\frac{y^{a-1}(a - by)}{e^{by}})$$

$$= (a^2 - a)y^{a-2} - aby^{a-1}$$

$$= \frac{y^{a-2}(a^2 - a - aby)e^{by} - by^{a-1}(a - by)e^{by}}{e^{2by}}$$

$$= \frac{y^{a-2}(a^2 - a - 2aby + b^2y^2)}{e^{by}}$$

At $y = 0$, $f'(0) = 0$. Without further investigation, we cannot determine if this point reflects a relative maximum, relative minimum, or neither.

At $y = \frac{a}{b}$:

$$\frac{d^2}{dy^2}(a/b) = \frac{(\frac{a}{b})^{a-2}(a^2 - a - 2ab\frac{a}{b} + b^2\frac{a^2}{b^2})}{e^{b\frac{a}{b}}}$$

$$= \frac{(\frac{a}{b})^{a-2}(-a)}{e^a}$$

$$= \frac{-a^{a-1}}{b^{a-2}e^a}$$

The sign of $f''(\frac{a}{b})$ could be positive, negative, or zero depending on the values of the constants a and b. Without this additional information, we cannot determine if the function has a relative maximum, minimum, or neither at critical point $\frac{a}{b}$.

On the other hand, if we confine a and b to positive real numbers, then $\frac{d^2}{dy^2}(y)$ is negative and we have a relative maximum.

Show also that $f(y)$ approaches zero as y tends to infinity

I'll provide an informal and intuitive solution:

$$\lim_{y\to\infty} f(y) = \lim_{y\to\infty} y^a/e^{by}$$

$$= \lim_{y\to\infty} \frac{e^{a\ lny}}{e^{by}}$$

Let's look at the exponents in the numerator and the denominator:

we know that that lny in the numerator grows more slowly than y in the denominator as y increases. If we assume b is positive, then the limit approaches zero as y becomes arbitrarily large. However, if b is negative, then the denominator approaches zero (but remains positive) as y tends to infinity. In this latter case, the limit of the function approaches infinity.

Page 566 Question 1

Use Euler's method to solve the first-order system subject to the specified initial condition. Use the given step size Δt and calculate the first three approximations (x_1, y_1), (x_2, y_2), and (x_3, y_3). Then repeat your calculations for $\frac{\Delta t}{2}$. Compare your approximations with the values of the given analytical solution.

Given:

$$\frac{dx}{dt} = 2x + 3y$$

$$\frac{dy}{dt} = 3x + 2y$$

$$x(0) = 1, y(0) = 1, \Delta t = \frac{1}{4}$$

Provided Analytical solution:

$$x(t) = \frac{1}{2}e^{-t} + \frac{1}{2}e^{5t},\ y(t) = \frac{1}{2}e^{-t} + \frac{1}{2}e^{5t}$$

First, we'll set up the problem in R:

```
# initial conditions
x0 <- 1
y0 <- 1
dt <- 0.25
dt_half <- dt/2
t0 = 0

# dxdt
dxdt <- function(x,y) 2*x + 3*y

# dydt
dydt <- function(x,y) 3*x + 2*y

# analytical solution x
x.t <- function(t){
  0.5 * exp(-t) + 0.5*exp(5*t)
}

# analytical solution y
y.t <- function(t){
  0.5 * exp(-t) + 0.5*exp(5*t)
}
```

Note: Euler's method employs the following recursive calculations:

$$x_i = x_{i-1} + f(t_{i-1}, x_{i-1}, y_{i-1})\Delta t$$

$$y_i = y_{i-1} + g(t_{i-1}, x_{i-1}, y_{i-1})\Delta t$$

Let's generate the first three approximations using step size $\frac{1}{4}$. We'll then compare the results to the analytical solutions.

```
### approximation, step size 1/4 ####

# x1,y1
x1 <- x0 + dxdt(x0,x0)*dt
y1 <- y0 + dydt(x0,y0)*dt

# x2, y2
x2 <- x1 + dxdt(x1,x1)*dt
y2 <- y1 + dydt(x1,y1)*dt

# x3, y3
x3 <- x2 + dxdt(x2,x2)*dt
y3 <- y2 + dydt(x2,y2)*dt

# ordered pair approximations, step size 1/4, saved as vector
```

```r
approx1 <- c(paste0("(",round(x1,4),",",round(y1,4),")"),
             paste0("(",round(x2,4),",",round(y2,4),")"),
             paste0("(",round(x3,4),",",round(y3,4),")"))

##### analytical results ####

# x1, y1
x1_act <- x.t(t0 + dt)
y1_act <- y.t(t0 + dt)

# x2, y2
x2_act <- x.t(t0 + dt*2)
y2_act <- y.t(t0 + dt*2)

# x3, y3
x3_act <- x.t(t0 + dt*3)
y3_act <- y.t(t0 + dt*3)

# save actual results to vector
actual <- c(paste0("(",round(x1_act,4),",",round(y1_act,4),")"),
            paste0("(",round(x2_act,4),",",round(y2_act,4),")"),
            paste0("(",round(x3_act,4),",",round(y3_act,4),")"))

# first three time steps
t <- c(0.25,0.50,0.75)

# save to df
df <- data.frame(t = t, approx=approx1, actual=actual)

# print
knitr::kable(df, align="rrr")
```

t	approx	actual
0.25	(2.25,2.25)	(2.1346,2.1346)
0.50	(5.0625,5.0625)	(6.3945,6.3945)
0.75	(11.3906,11.3906)	(21.4967,21.4967)

Finally, we'll generate the first three approximations using step size $\frac{1}{8}$. Once again, we'll then compare the results to the analytical solutions.

```r
## step size 1/8 ####

# x1,y1
x1_half <- x0 + dxdt(x0,x0)*dt_half
y1_half <- y0 + dydt(x0,y0)*dt_half

# x2, y2
x2_half <- x1_half + dxdt(x1_half,x1_half)*dt_half
y2_half <- y1_half + dydt(x1_half,y1_half)*dt_half

# x3, y3
x3_half <- x2_half + dxdt(x2_half,x2_half)*dt_half
```

```r
y3_half <- y2_half + dydt(x2_half,y2_half)*dt_half

# ordered pair approximations, step size 1/4, saved as vector
approx2 <- c(paste0("(",round(x1_half,4),",",round(y1_half,4),")"),
             paste0("(",round(x2_half,4),",",round(y2_half,4),")"),
             paste0("(",round(x3_half,4),",",round(y3_half,4),")"))

##### analytical results ####

# x1, y1
x1_act_half <- x.t(t0 + dt_half)
y1_act_half <- y.t(t0 + dt_half)

# x2, y2
x2_act_half <- x.t(t0 + dt_half*2)
y2_act_half <- y.t(t0 + dt_half*2)

# x3, y3
x3_act_half <- x.t(t0 + dt_half*3)
y3_act_half <- y.t(t0 + dt_half*3)

# save analytical results to vector
actual_half <- c(paste0("(",round(x1_act_half,4),",",round(y1_act_half,4),")"),
            paste0("(",round(x2_act_half,4),",",round(y2_act_half,4),")"),
            paste0("(",round(x3_act_half,4),",",round(y3_act_half,4),")"))

# first three time periods
t <- c(1/8,2/8,3/8)

# save to df
df <- data.frame(t = t, approx=approx2, actual=actual_half)

# print
knitr::kable(df, align="rrr")
```

t	approx	actual
0.125	(1.625,1.625)	(1.3754,1.3754)
0.250	(2.6406,2.6406)	(2.1346,2.1346)
0.375	(4.291,4.291)	(3.6041,3.6041)