Numerical Analysis HW 7

Michael Vu

August 31, 2023

Problem 1. We are asked to solve problem 7 of section 5.6. Below is the ODE driver for solving the problem:

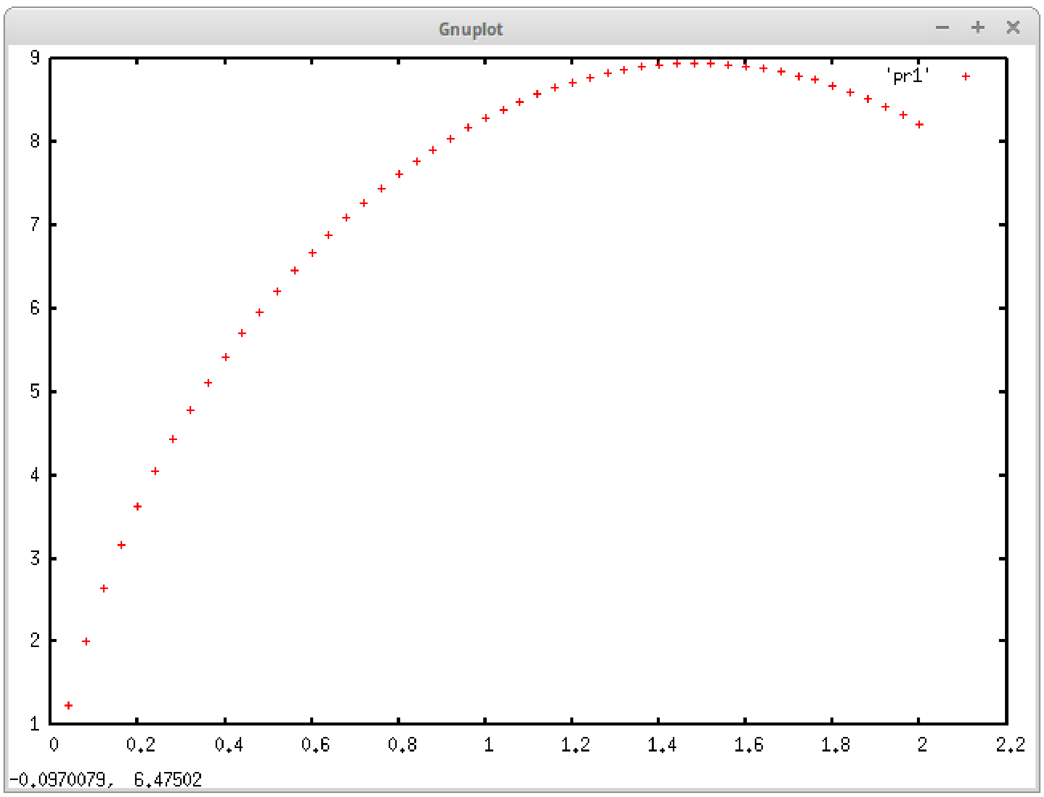
```
implicit none
double precision :: t,tout,relerr,abserr,a,b,dt
double precision, allocatable, dimension(:) :: y(:), work(:)
integer, allocatable, dimension(:) :: iwork(:)
integer :: nsteps,i,iflag,neqn
external f

neqn = 1

allocate( y(neqn), work(100+21*neqn), iwork(5) )

y = (/ 0.0d0 /)
a = 0.0d0
b = 2.0d0
nsteps = 100
t = a
dt = (b-a)/dble(nsteps)
relerr = 1.0d-10
abserr = 1.0d-10
iflag = 1

do i = 1,nsteps
tout = t + dt
call ode(f,neqn,y,t,tout,relerr,abserr,iflag,work,iwork)
```

```
if(iflag /= 2) then
print*, 'Warning: i, iflag = ', i, iflag
end if

print*, tout, y(1)
end do

deallocate(y,work,iwork)
stop
end

subroutine f(t,y,yp)
implicit none
double precision :: t,y(1),yp(1)

yp(1) = (110.d0 * cos(t) - y(1)) / (1.1d0 * (2.1d0 + 1.8d0 * y(1)))


return
end
```

These are the first and last few interations leading up to $t = 2$:

```
t=                        i(t)=
2.0000000000000000E-002   0.72417563949512709
4.0000000000000001E-002   1.2367583733012615
5.9999999999999998E-002   1.6558366962986306
.
.
.
1.9600000000000013        8.3226127553865492
1.9800000000000013        8.2681151148920353
2.0000000000000013        8.2111871600797919
```

So we find that $i(2)$ is approximately 8.211871600797919. Below is the graph as we travel forward in time from 0 to 2:



Gnuplot
'pr1'
9
8
7
6
5
4
3
2
1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
2.2
-0.0970079, 6.47502

Problem 2. We are asked to solve problem 5 of section 5.5. Below is the ODE driver to solve the problem:

```
implicit none
double precision :: t,tout,relerr,abserr,a,b,dt
double precision, allocatable, dimension(:) :: y(:), work(:)
integer, allocatable, dimension(:) :: iwork(:)
integer :: nsteps,i,iflag,neqn
external f

neqn = 1

allocate( y(neqn), work(100+21*neqn), iwork(5) )

y = (/ 50976.0d0 /)
a = 0.0d0
b = 5.0d0
nsteps = 100
t = a
dt = (b-a)/dble(nsteps)
relerr = 1.0d-10
abserr = 1.0d-10
iflag = 1

do i = 1,nsteps
tout = t + dt
call ode(f,neqn,y,t,tout,relerr,abserr,iflag,work,iwork)

print*, tout, y(1)

if(iflag /= 2) then
print*, 'Warning: i, iflag = ', i, iflag
stop
end if
end do

deallocate(y,work,iwork)
stop
```

```
end

subroutine f(t,y,yp)
implicit none
double precision :: t,y(1),yp(1)

yp(1) = 2.9d-2 * y(1) - 1.4d-7 * (y(1)**2)

return
end
```

These are the first and last few interations leading up to $t = 5$:

```
t=                        P(t)=
5.0000000000000003E-002   51031.745846037498
0.10000000000000001       51087.532711320688
0.15000000000000002       51143.360582430090
.
.
.
4.8999999999999906        56631.626639862254
4.9499999999999904        56691.312111451043
4.9999999999999902        56751.036768758218
```

So we find that $P(5)$ is approximately 56751.036768758218.

Problem 3. In this problem we are asked to solve an initial value problem given by Dr. Glunt, this is the ODE driver use to solve this problem:

```
implicit none
double precision :: t,tout,relerr,abserr,a,b,dt
double precision, allocatable, dimension(:) :: y(:), work(:)
integer, allocatable, dimension(:) :: iwork(:)
integer :: nsteps,i,iflag,neqn
external f

neqn = 2

allocate( y(neqn), work(100+21*neqn), iwork(5) )
```

```
y = (/ 1.0d0,0.0d0 /)
a = 0.0d0
b = 10.0d0
nsteps = 100
t = a
dt = (b-a)/dble(nsteps)
relerr = 1.0d-10
abserr = 1.0d-10
iflag = 1

do i = 1,nsteps
tout = t + dt
call ode(f,neqn,y,t,tout,relerr,abserr,iflag,work,iwork)

print*, t, y(1)

if(iflag /= 2) then
print*, 'Warning: i, iflag = ', i, iflag
stop
end if
end do

deallocate(y,work,iwork)
stop
end

subroutine f(t,y,yp)
implicit none
double precision :: t,y(2),yp(2)

yp(1) = y(2)
yp(2) = -1.0d0 / (y(1)**2)

return
end
```

This is the output from the code:

```
t=                          z(t)=
0.10000000000000001         0.99499163596971130
0.20000000000000001         0.97986467324765869
0.30000000000000004         0.95430172599537555
0.40000000000000002         0.91773111353375358
0.50000000000000000         0.86924869765427382
0.59999999999999998         0.80747099501982766
0.69999999999999996         0.73024958947622853
0.79999999999999993         0.63405965896115235
0.89999999999999991         0.51244965313027724
0.99999999999999989         0.35068159575822078
1.0999999999999999          7.8972465069453615E-002
1.1107207348315986          2.7160802790846351E-008
Warning: i, iflag =  12, 4
```

It appears to have had an error when running the program. Since this problem sets up a scenario where two stars are on a collision path, the point in time which we receive this error indicates a collision as occured.

Problem 4. This problem is similar to problem 3. The only change that was made was to halt the program when $z \leq 0.01$. Problem 3 sets up a collision between the stars as point particles rather than an object that actually takes some space. Setting up the problem in this new manner should simulate a better scenario of two stars colliding. This is the ODE driver used to solve this problem:

```
implicit none
double precision :: t,tout,relerr,abserr,a,b,dt
double precision, allocatable, dimension(:) :: y(:), work(:)
integer, allocatable, dimension(:) :: iwork(:)
integer :: nsteps,i,iflag,neqn
external f

neqn = 2

allocate( y(neqn), work(100+21*neqn), iwork(5) )

y = (/ 1.0d0,0.0d0 /)
```

```
a = 0.0d0
b = 10.0d0
nsteps = 100
t = a
dt = (b-a)/dble(nsteps)
relerr = 1.0d-10
abserr = 1.0d-10
iflag = 1

do i = 1,nsteps
tout = t + dt
call ode(f,neqn,y,t,tout,relerr,abserr,iflag,work,iwork)

print*, t, y(1)

if(y(1) <= 0.01d0) then
print*, 'Collision has occured'
stop
end if

if(iflag /= 2) then
print*, 'Warning: i, iflag = ', i, iflag
stop
end if
end do

deallocate(y,work,iwork)
stop
end

subroutine f(t,y,yp)
implicit none
double precision :: t,y(2),yp(2)

yp(1) = y(2)
yp(2) = -1.0d0 / (y(1)**2)

return
```

```
end
```

This is the output from the code:

```
t=                        z(t)=
0.10000000000000001       0.99499163596971130
0.20000000000000001       0.97986467324765869
0.30000000000000004       0.95430172599537555
0.40000000000000002       0.91773111353375358
0.50000000000000000       0.86924869765427382
0.59999999999999998       0.80747099501982766
0.69999999999999996       0.73024958947622853
0.79999999999999993       0.63405965896115235
0.89999999999999991       0.51244965313027724
0.99999999999999989       0.35068159575822078
1.0999999999999999        7.8972465069453615E-002
1.1107207348315986        2.7160802790846351E-008
Collision has occured
```

This should have resulted in different stopping times, however, it did not. This is either an error on my part, or something else. In either case, the output stopped when $z(t = 1.1107207348315986) = 2.7160802790846351 \times 10^{-8}$