Pendulum in cartesian coordinates

Taylor based numerical scheme

```
> restart:
```

Pendulum equation

```
> EQ1 := mass*diff(x(t),t,t)+2*x(t)*lambda(t) ;
  EQ2 := mass*diff(y(t),t,t)+2*y(t)*lambda(t)+mass*g ;
  EQ3 := x(t)^2+y(t)^2-1 ;
```

$$EQ1 := mass\left(\frac{d^2}{dt^2}x(t)\right) + 2\,x(t)\,\lambda(t)$$

$$EQ2 := mass\left(\frac{d^2}{dt^2}y(t)\right) + 2\,y(t)\,\lambda(t) + mass\,g$$

$$EQ3 := x(t)^2 + y(t)^2 - 1 \qquad (1)$$

Derivate constraint two times

```
> DEQ3  := diff(EQ3,t);
  DDEQ3 := diff(DEQ3,t);
```

$$DEQ3 := 2\,x(t)\left(\frac{d}{dt}x(t)\right) + 2\,y(t)\left(\frac{d}{dt}y(t)\right)$$

$$DDEQ3 := 2\left(\frac{d}{dt}x(t)\right)^2 + 2\,x(t)\left(\frac{d^2}{dt^2}x(t)\right) + 2\left(\frac{d}{dt}y(t)\right)^2 + 2\,y(t)\left(\frac{d^2}{dt^2}y(t)\right) \qquad (2)$$

Solve for second derivative

```
> RES := solve( {EQ1,EQ2,DDEQ3}, diff({x(t),y(t)},t,t) union {lambda
  (t)} ) ;
```

$$RES := \left\{\frac{d^2}{dt^2}x(t) = \frac{x(t)\left(y(t)\,g - \left(\frac{d}{dt}x(t)\right)^2 - \left(\frac{d}{dt}y(t)\right)^2\right)}{x(t)^2 + y(t)^2}, \frac{d^2}{dt^2}y(t) = -\frac{x(t)^2\,g + y(t)\left(\frac{d}{dt}x(t)\right)^2 + y(t)\left(\frac{d}{dt}y(t)\right)^2}{x(t)^2 + y(t)^2}, \lambda(t) = -\frac{1}{2}\,\frac{mass\left(y(t)\,g - \left(\frac{d}{dt}x(t)\right)^2 - \left(\frac{d}{dt}y(t)\right)^2\right)}{x(t)^2 + y(t)^2}\right\} \qquad (3)$$

Change names

```
> SUBS := { diff(x(t),t,t) = ax(t),
            diff(y(t),t,t) = ay(t),
            diff(x(t),t)   = u(t),
            diff(y(t),t)   = v(t) } ;
```

(4)

$$SUBS := \left\{ \frac{d}{dt} x(t) = u(t), \frac{d^2}{dt^2} x(t) = ax(t), \frac{d}{dt} y(t) = v(t), \frac{d^2}{dt^2} y(t) = ay(t) \right\} \quad (4)$$

```
> subs(SUBS,RES) ;
```

$$\left\{ ax(t) = \frac{x(t)\left(y(t)\, g - u(t)^2 - v(t)^2\right)}{x(t)^2 + y(t)^2}, ay(t) = -\frac{x(t)^2 g + y(t)\, u(t)^2 + y(t)\, v(t)^2}{x(t)^2 + y(t)^2}, \lambda(t) = -\frac{1}{2} \frac{mass\left(y(t)\, g - u(t)^2 - v(t)^2\right)}{x(t)^2 + y(t)^2} \right\} \quad (5)$$

Advancing with Taylor

```
> XKP1 := x(t)+u(t)*DT+ax(t)*DT^2/2 ;
  YKP1 := y(t)+v(t)*DT+ay(t)*DT^2/2 ;
  UKP1 := u(t)+ax(t)*DT ;
  VKP1 := v(t)+ay(t)*DT ;
```

$$XKP1 := x(t) + u(t)\, DT + \frac{1}{2} ax(t)\, DT^2$$

$$YKP1 := y(t) + v(t)\, DT + \frac{1}{2} ay(t)\, DT^2$$

$$UKP1 := u(t) + ax(t)\, DT$$

$$VKP1 := v(t) + ay(t)\, DT \quad (6)$$

Substituting acceleration

```
> XKP1 := subs( subs(SUBS,RES), XKP1) ;
  YKP1 := subs( subs(SUBS,RES), YKP1) ;
  UKP1 := subs( subs(SUBS,RES), UKP1) ;
  VKP1 := subs( subs(SUBS,RES), VKP1) ;
```

$$XKP1 := x(t) + u(t)\, DT + \frac{1}{2} \frac{x(t)\left(y(t)\, g - u(t)^2 - v(t)^2\right) DT^2}{x(t)^2 + y(t)^2}$$

$$YKP1 := y(t) + v(t)\, DT - \frac{1}{2} \frac{\left(x(t)^2 g + y(t)\, u(t)^2 + y(t)\, v(t)^2\right) DT^2}{x(t)^2 + y(t)^2}$$

$$UKP1 := u(t) + \frac{x(t)\left(y(t)\, g - u(t)^2 - v(t)^2\right) DT}{x(t)^2 + y(t)^2}$$

$$VKP1 := v(t) - \frac{\left(x(t)^2 g + y(t)\, u(t)^2 + y(t)\, v(t)^2\right) DT}{x(t)^2 + y(t)^2} \quad (7)$$

Build numerical scheme

```
> SUBSV := { x(t)=xO, y(t)=yO, u(t)=uO, v(t)=vO, mu(t)=muN } ;
```

$$SUBSV := \{\mu(t) = muN, u(t) = uO, v(t) = vO, x(t) = xO, y(t) = yO\} \quad (8)$$

```
> XKP1 := subs(SUBSV,XKP1) ;
  YKP1 := subs(SUBSV,YKP1) ;
  UKP1 := subs(SUBSV,UKP1) ;
  VKP1 := subs(SUBSV,VKP1) ;
```

$$XKP1 := xO + uO\, DT + \frac{1}{2} \frac{xO\left(g\, yO - uO^2 - vO^2\right) DT^2}{xO^2 + yO^2}$$

$$YKP1 := yO + vO\,DT - \frac{1}{2}\,\frac{\left(g\,xO^2 + uO^2\,yO + vO^2\,yO\right)DT^2}{xO^2 + yO^2}$$

$$UKP1 := uO + \frac{xO\left(g\,yO - uO^2 - vO^2\right)DT}{xO^2 + yO^2}$$

$$VKP1 := vO - \frac{\left(g\,xO^2 + uO^2\,yO + vO^2\,yO\right)DT}{xO^2 + yO^2} \tag{9}$$

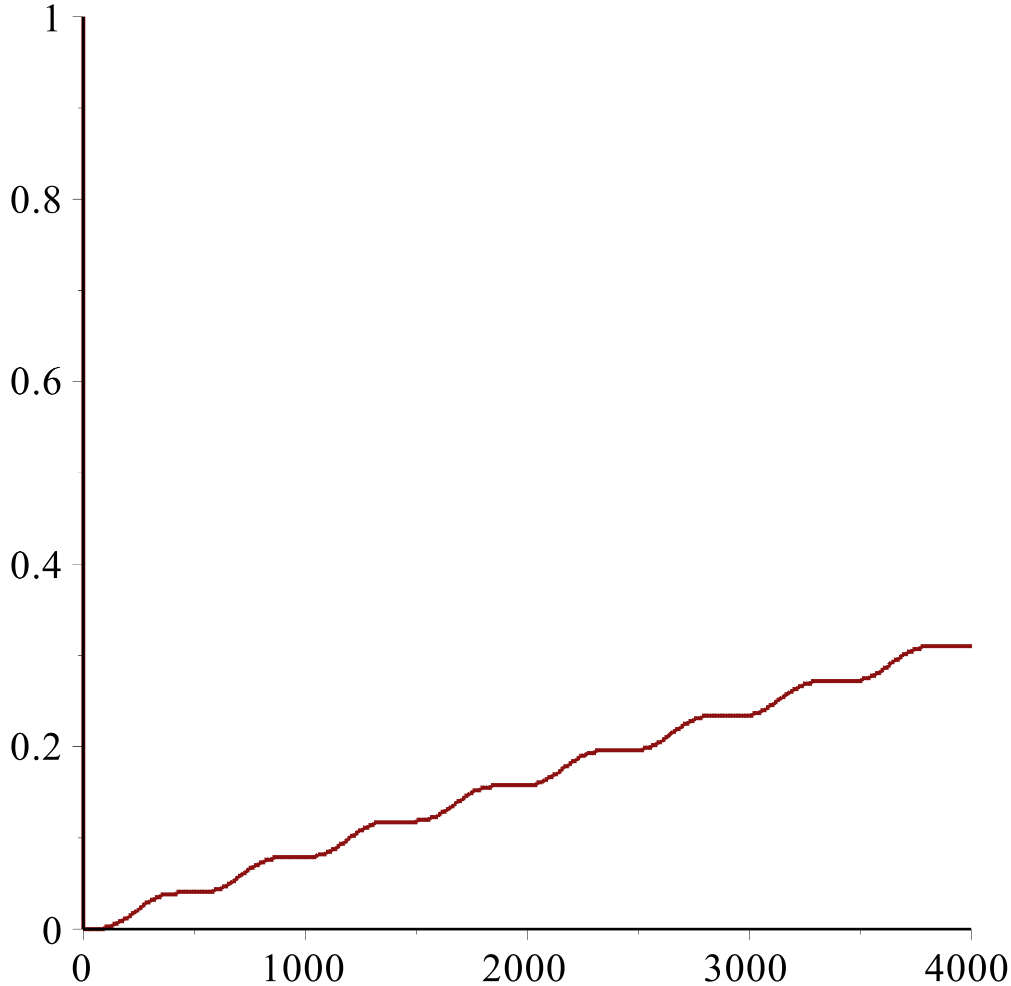
```
> advance := proc ( x0, y0, u0, v0, dt, N )
    local kk, SUBS, x1, y1, u1, v1, XY,UV,R ;
    XY := [[x0,y0]] ;
    UV := [[u0,v0]] ;
    R  := [[0,1]] ;
    for kk from 1 to N do
      SUBS := { g=9.81, DT=dt,
                xO=XY[-1][1], yO=XY[-1][2],
                uO=UV[-1][1], vO=UV[-1][2]} ;
      x1  := evalf(subs( SUBS, XKP1 )) ;
      y1  := evalf(subs( SUBS, YKP1 )) ;
      u1  := evalf(subs( SUBS, UKP1 )) ;
      v1  := evalf(subs( SUBS, VKP1 )) ;
      XY  := [op(XY),[x1,y1]] ;
      UV  := [op(UV),[u1,v1]] ;
      R   := [op(R), [kk,x1^2+y1^2-1]] ;
    end  ;
    [XY,UV,R] ;
  end proc:
```

Test numerical scheme DT = 1/200

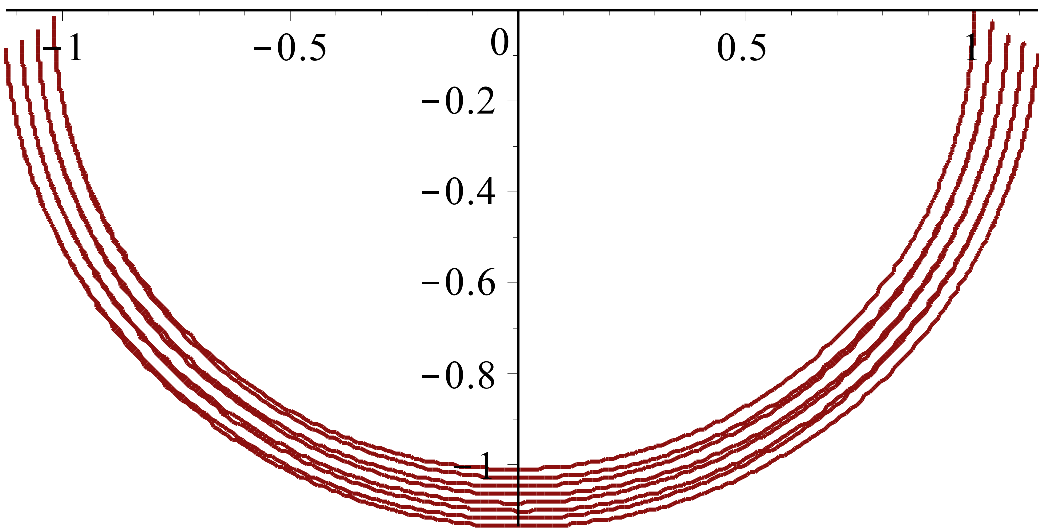
```
> 10/1000.0 ;
```

$$0.01000000000 \tag{10}$$

```
> RES := advance( 1, 0, 0, 0, 10/4000, 4000 ) :
> plot( RES[3] ) ;
```

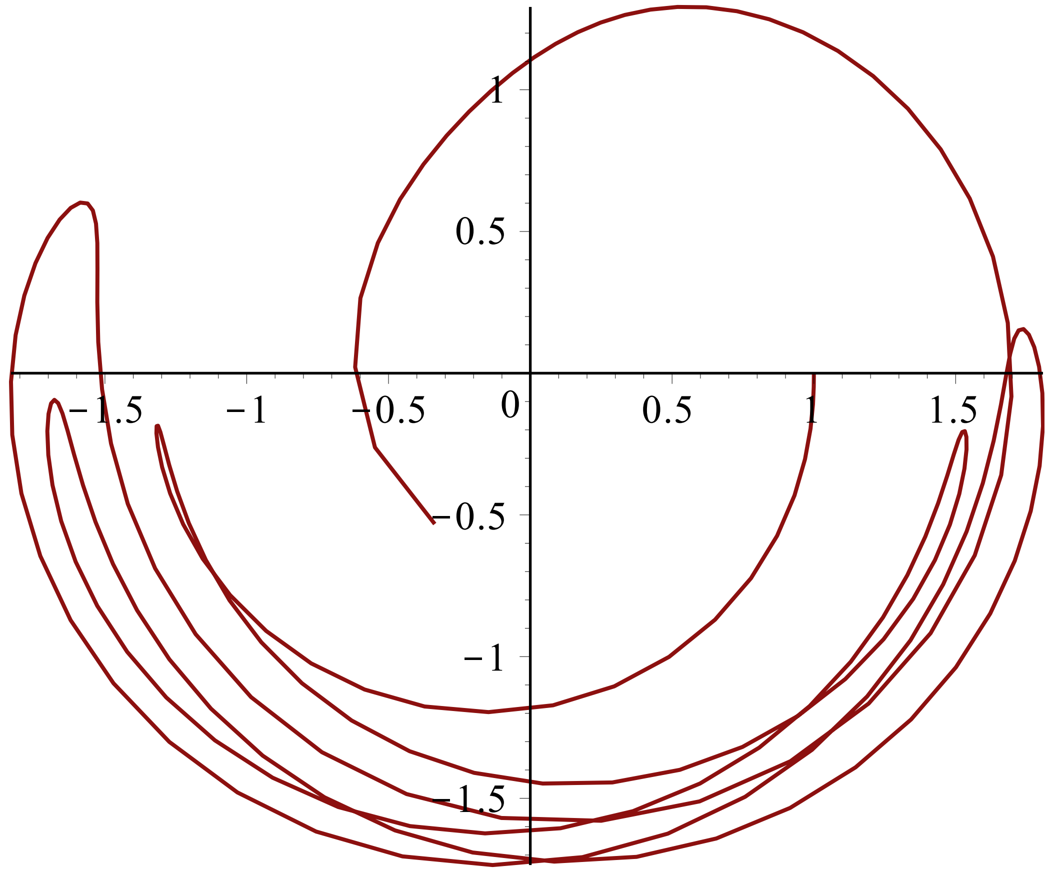



```
> plot( RES[1], scaling=constrained ) ;
```

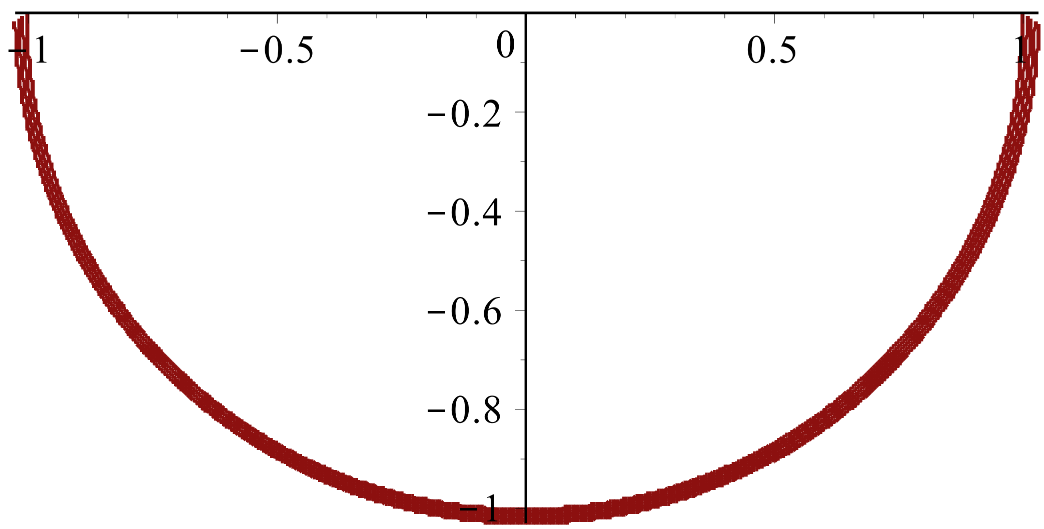


Test numerical scheme DT = 1/20

```
> RES := advance( 1, 0, 0, 0, 10/200, 200 ) :
  plot( RES[1], scaling=constrained ) ;
```




```
> RES := advance( 1, 0, 0, 0, 10/20000, 20000 ) :
  plot( RES[1], scaling=constrained ) ;
```



-1
-0.5
0
0.5
1
-0.2
-0.4
-0.6
-0.8
-1