Solve a DAE using non stabilized numerical schemes

```
> restart:
> with(plots):
```

The DAE

```
> EQ1 := diff(y(t),t)-q1+x(t) ;
  EQ2 := diff(z(t),t)-q2+(1+eta)*y(t)+eta*t*(q1-x(t)) ;
  EQ3 := q3-eta*t*y(t)-z(t) ;
```

$$EQ1 := \frac{d}{dt} y(t) - q1 + x(t)$$

$$EQ2 := \frac{d}{dt} z(t) - q2 + (1+\eta)\, y(t) + \eta\, t\, (q1 - x(t))$$

$$EQ3 := q3 - \eta\, t\, y(t) - z(t) \qquad (1)$$

```
> EXACT := x(t)=cos(t),y(t)=sin(t),z(t)=t^2;
```

$$EXACT := x(t) = \cos(t), y(t) = \sin(t), z(t) = t^2 \qquad (2)$$

```
> expand(subs(EXACT,[EQ1,EQ2,EQ3] )) ;
  SOLQ := op(solve( %, [q1,q2,q3] )) ;
```

$$[2\cos(t) - q1, 2\,t - q2 + \sin(t) + \sin(t)\,\eta + \eta\, t\, q1 - \eta\, t\cos(t), q3 - \eta\, t\sin(t) - t^2]$$

$$SOLQ := [q1 = 2\cos(t), q2 = 2\,t + \sin(t) + \sin(t)\,\eta + \eta\, t\cos(t), q3 = \eta\, t\sin(t) + t^2] \qquad (3)$$

```
> DAE := simplify(subs(eta=-1,subs( SOLQ, [EQ1,EQ2,EQ3] ))): <%> ;
```

$$\begin{bmatrix} \frac{d}{dt} y(t) - 2\cos(t) + x(t) \\ \frac{d}{dt} z(t) - 2\,t - t\cos(t) + t\,x(t) \\ -t\sin(t) + t^2 + t\,y(t) - z(t) \end{bmatrix} \qquad (4)$$

Compute the index and transform the DAE to an ODE

```
> SOLYZ := solve( DAE[1..2], diff({y(t),z(t)},t) ) ;
```

$$SOLYZ := \left\{ \frac{d}{dt} y(t) = 2\cos(t) - x(t), \frac{d}{dt} z(t) = 2\,t + t\cos(t) - t\,x(t) \right\} \qquad (5)$$

Compute the first hidden constraint

```
> HIDDEN1 := subs( SOLYZ, diff(DAE[3],t) ) ;
```

$$HIDDEN1 := -\sin(t) - 2\,t\cos(t) + y(t) + t\,(2\cos(t) - x(t)) + t\,x(t) \qquad (6)$$

Compute the second hidden constraint

```
> HIDDEN2 := subs( SOLYZ, simplify(diff(HIDDEN1,t)) ) ;
```

$$HIDDEN2 := \cos(t) - x(t) \qquad (7)$$

```
> DAE3 := subs( SOLYZ, simplify(diff(HIDDEN2,t)) ) ;
```

$$DAE3 := -\sin(t) - \left(\frac{d}{dt} x(t)\right) \qquad (8)$$

The DAE has index 3 and after index reduction is the following ODE

```
> ODE := [-DAE3,op(DAE[1..2])];
```

$$ODE := \left[\sin(t) + \frac{d}{dt}x(t), \frac{d}{dt}y(t) - 2\cos(t) + x(t), \frac{d}{dt}z(t) - 2t - t\cos(t) + t\,x(t)\right] \quad (9)$$

Approximate solution of the ODE by using Explicit Euler

```
> SUBS := diff(x(t), t) = (x[k+1]-x[k])/h,
          diff(y(t), t) = (y[k+1]-y[k])/h,
          diff(z(t), t) = (z[k+1]-z[k])/h,
          x(t)    = x[k],
          y(t)    = y[k],
          z(t)    = z[k],
          t       = k*h ;
```

$$SUBS := \frac{d}{dt}x(t) = \frac{x_{k+1} - x_k}{h}, \frac{d}{dt}y(t) = \frac{y_{k+1} - y_k}{h}, \frac{d}{dt}z(t) = \frac{z_{k+1} - z_k}{h}, x(t) = x_k, y(t) = y_k, z(t) = z_k, t = k\,h \quad (1.1)$$

```
> ExplicitEuler := subs(SUBS,ODE) ;
```

$$ExplicitEuler := \left[\sin(k\,h) + \frac{x_{k+1} - x_k}{h}, \frac{y_{k+1} - y_k}{h} - 2\cos(k\,h) + x_k, \frac{z_{k+1} - z_k}{h} - 2\,k\,h - k\,h\cos(k\,h) + k\,h\,x_k\right] \quad (1.2)$$

```
> ExplicitEulerAdvance := solve( ExplicitEuler, {x[k+1],y[k+1],z[k+1]} ): <op(%)> ;
```

$$\begin{bmatrix} x_{k+1} = -\sin(k\,h)\,h + x_k \\ y_{k+1} = y_k + 2\cos(k\,h)\,h - x_k\,h \\ z_{k+1} = -k\,h^2\,x_k + z_k + 2\,k\,h^2 + k\,h^2\cos(k\,h) \end{bmatrix} \quad (1.3)$$

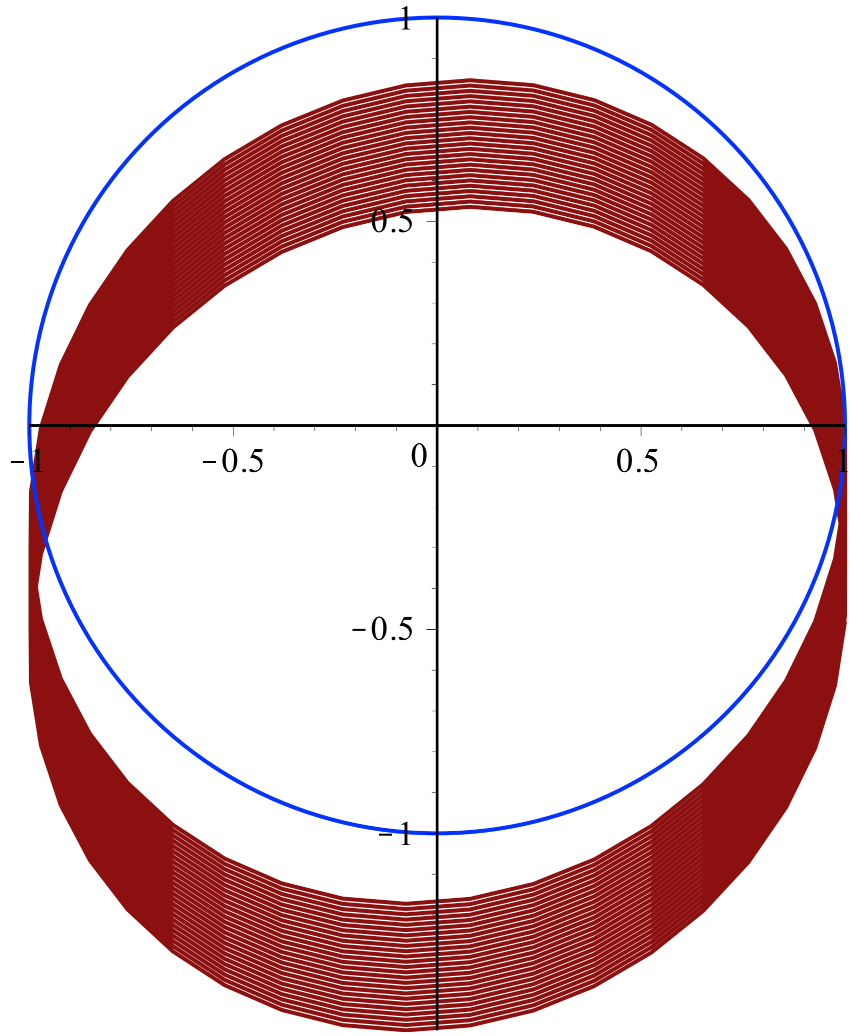
```
> T := 50*Pi ;
  N := 1000 ;
  h := T/N ;
```

$$T := 50\,\pi$$

$$N := 1000$$

$$h := \frac{1}{20}\,\pi \quad (1.4)$$

```
> X := [1] :
  Y := [0] :
  Z := [0] :
  for k from 1 to N do
    SOLSTEP := subs(x[k]=X[-1],y[k]=Y[-1],z[k]=Z[-1],
  ExplicitEulerAdvance):
    X := [op(X),evalf(subs( SOLSTEP, x[k+1] ))] ;
    Y := [op(Y),evalf(subs( SOLSTEP, y[k+1] ))] ;
    Z := [op(Z),evalf(subs( SOLSTEP, z[k+1] ))] ;
  end:
> A := plot( [seq([X[k],Y[k]],k=1..N)],scaling=CONSTRAINED):
  B := plot( [cos(t),sin(t),t=0..2*Pi],color=blue):
> display(A,B);
```




```
> DAE[3] ;
```

$$-t\sin(t)+t^2+t\,y(t)-z(t) \quad \textbf{(1.5)}$$

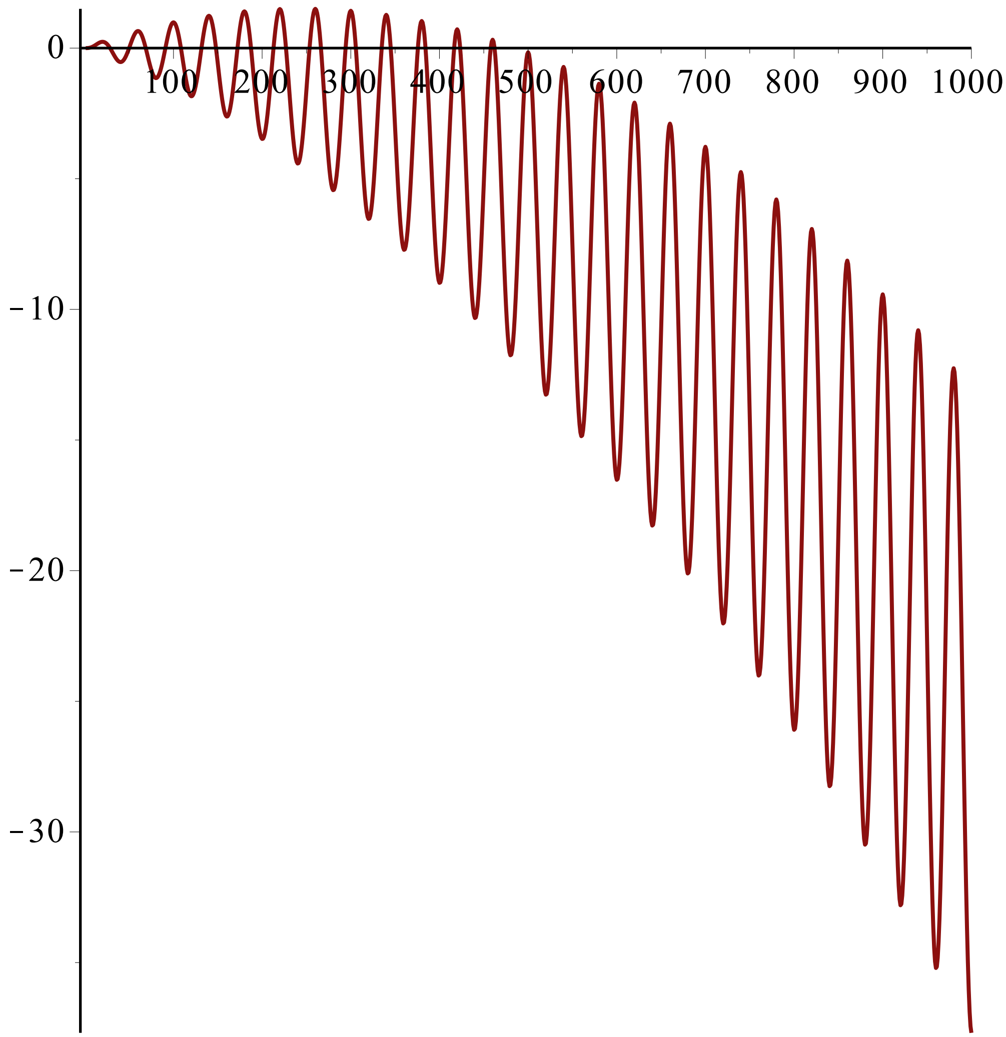
```
> C1 := Vector(N) ;
  C2 := Vector(N) ;
  C3 := Vector(N) ;
  for k from 1 to N do
    C1[k] := evalf(subs( x(t)=X[k],y(t)=Y[k], z(t)=Z[k], t=k*h, DAE
  [3])) :
    C2[k] := evalf(subs( x(t)=X[k],y(t)=Y[k], z(t)=Z[k], t=k*h,
  HIDDEN1)) :
    C3[k] := evalf(subs( x(t)=X[k],y(t)=Y[k], z(t)=Z[k], t=k*h,
  HIDDEN2)) :
  end:
```

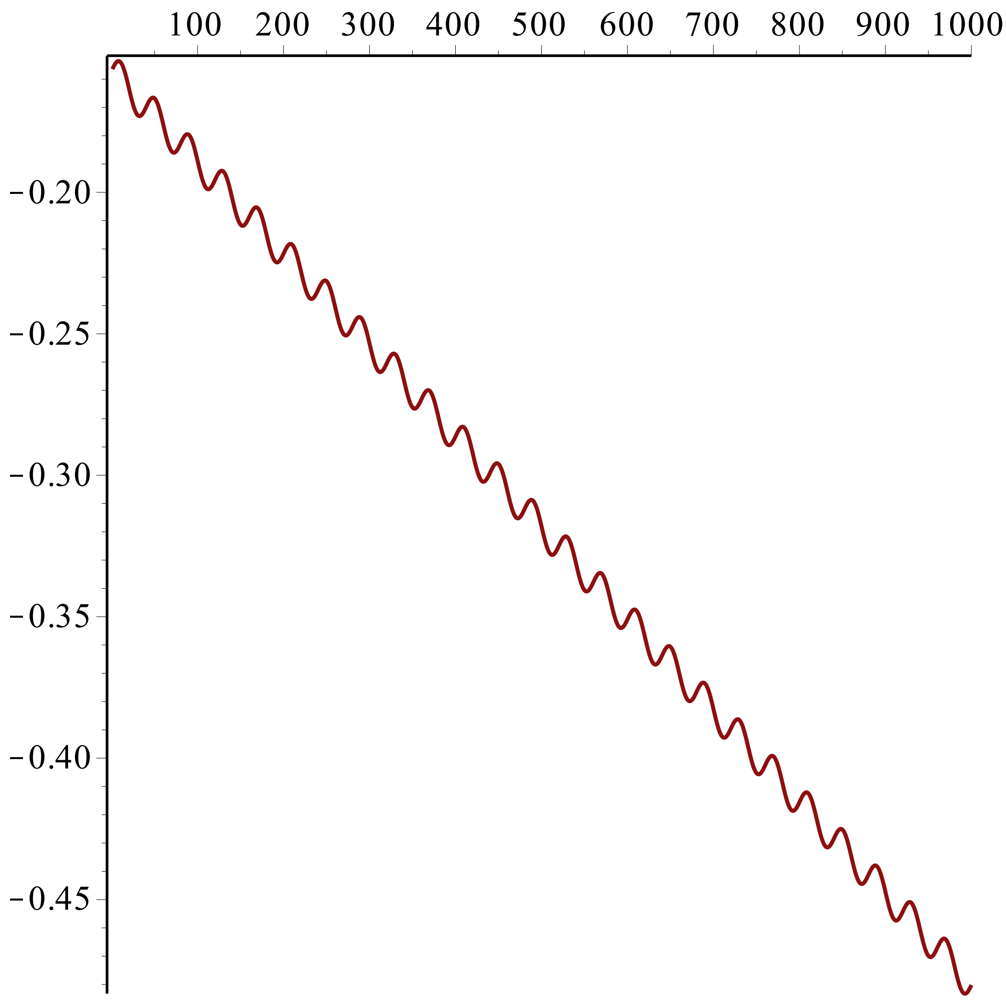
$$C1 := \begin{bmatrix} 1 \,..\, 1000 \text{ Vector}_{column} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

$$C2 := \begin{bmatrix} \text{1 .. 1000 } Vector_{column} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

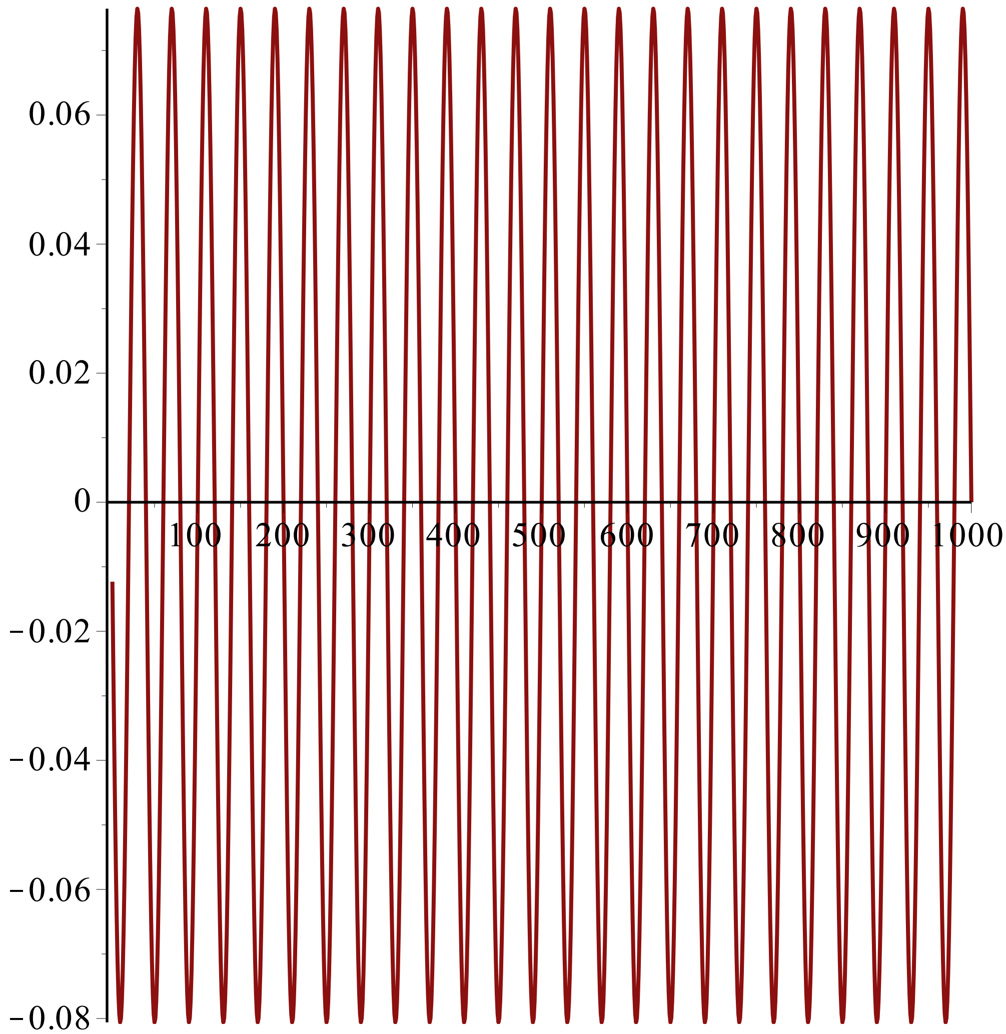
$$C3 := \begin{bmatrix} \text{1 .. 1000 } Vector_{column} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix} \qquad (1.6)$$

```
> plot( [seq([k,C1[k]],k=1..N)] ) ;
  plot( [seq([k,C2[k]],k=1..N)] ) ;
  plot( [seq([k,C3[k]],k=1..N)] ) ;
```





100
200
300
400
500
600
700
800
900
1000
−0.20
−0.25
−0.30
−0.35
−0.40
−0.45



0.06
0.04
0.02
0
100
200
300
400
500
600
700
800
900
1000
−0.02
−0.04
−0.06
−0.08