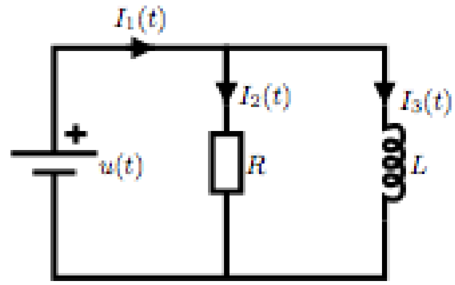
Example of numerical solution of a DAE




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

```
> EQ1 := L * diff(I3(t),t) - u(t) ;
  EQ2 := I1(t)-I2(t)-I3(t) ;
  EQ3 := u(t)-R*I2(t) ;
```

$$EQ1 := L\left(\frac{d}{dt} I3(t)\right) - u(t)$$

$$EQ2 := I1(t) - I2(t) - I3(t)$$

$$EQ3 := u(t) - R\, I2(t) \quad (1)$$

Index calculation

```
> DEQ2 := diff(EQ2,t) ;
  DEQ3 := diff(EQ3,t) ;
```

$$DEQ2 := \frac{d}{dt} I1(t) - \left(\frac{d}{dt} I2(t)\right) - \left(\frac{d}{dt} I3(t)\right)$$

$$DEQ3 := \frac{d}{dt} u(t) - R\left(\frac{d}{dt} I2(t)\right) \quad (2)$$

The DAE is of index 1

```
> ODE := solve( { EQ1, DEQ2, DEQ3} , diff({I1(t),I2(t),I3(t)},t) )
  ;
```

$$ODE := \left\{ \frac{d}{dt} I3(t) = \frac{u(t)}{L}, \frac{d}{dt} I1(t) = \frac{\left(\frac{d}{dt} u(t)\right) L + u(t)\, R}{L\, R}, \frac{d}{dt} I2(t) = \frac{\frac{d}{dt} u(t)}{R} \right\} \quad (3)$$

```
> SUBS := { L=0.001, R=1, u(t)=cos(100*t) } ;
```

$$SUBS := \{ L = 0.001, R = 1, u(t) = \cos(100\, t) \} \quad (4)$$

```
> SOL := dsolve( subs(SUBS,ODE) union { I1(0)=1, I2(0)=1, I3(0)=0
  } )  ;
```

$$SOL := \{ I1(t) = \cos(100\, t) + 10 \sin(100\, t), I3(t) = 10 \sin(100\, t), I2(t) = \cos(100\, t) \} \quad (5)$$

```
> I1_esatta := unapply( subs(SOL,I1(t)), t);
  I2_esatta := unapply( subs(SOL,I2(t)), t);
  I3_esatta := unapply( subs(SOL,I3(t)), t);
```

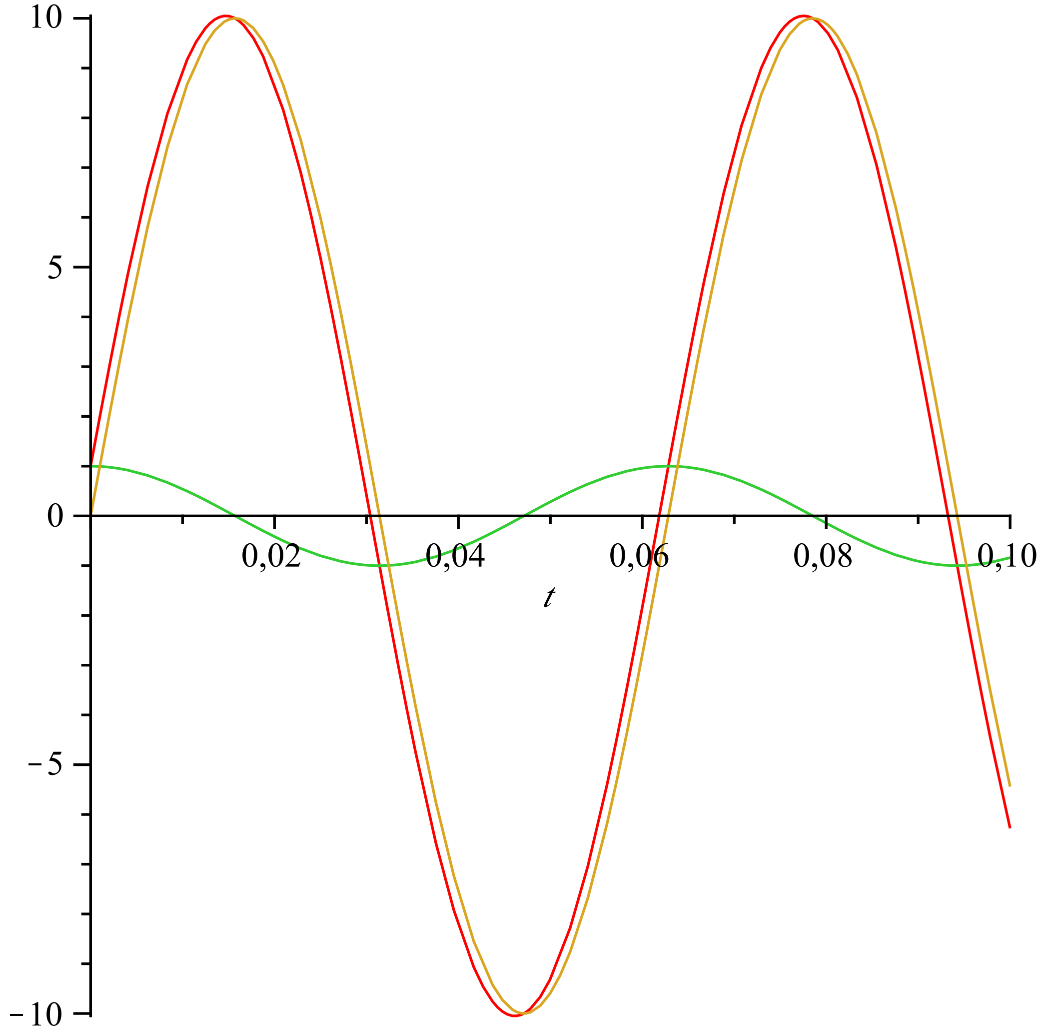
$$I1\_esatta := t \rightarrow \cos(100\, t) + 10 \sin(100\, t)$$

$$I2\_esatta := t \rightarrow \cos(100\, t)$$

$$I3\_esatta := t \rightarrow 10 \sin(100\, t) \quad (6)$$

```
> ESATTA := plot( subs(SOL,[I1(t),I2(t),I3(t)]), t=0..0.1 ):
```

```
> display(ESATTA) ;
```



Coodinate partitioning

Solve I1 and I2 respect to I3

```
> SOLI12 := solve( {EQ2,EQ3}, {I1(t),I2(t)}) ;
```

$$SOLI12 := \left\{ I1(t) = \frac{u(t) + I3(t)\,R}{R},\ I2(t) = \frac{u(t)}{R} \right\} \qquad (1.1)$$

```
> subs( SOLI12, EQ1) ;
```

$$L \left(\frac{d}{dt}\, I3(t) \right) - u(t) \qquad (1.2)$$

Solve numerically with Crank-Nicolson

```
> ALG := subs( I1(t)=I1_N, I2(t)=I2_N, I3(t)=I3_N, u(t)=u_N,
  SOLI12) ;
```

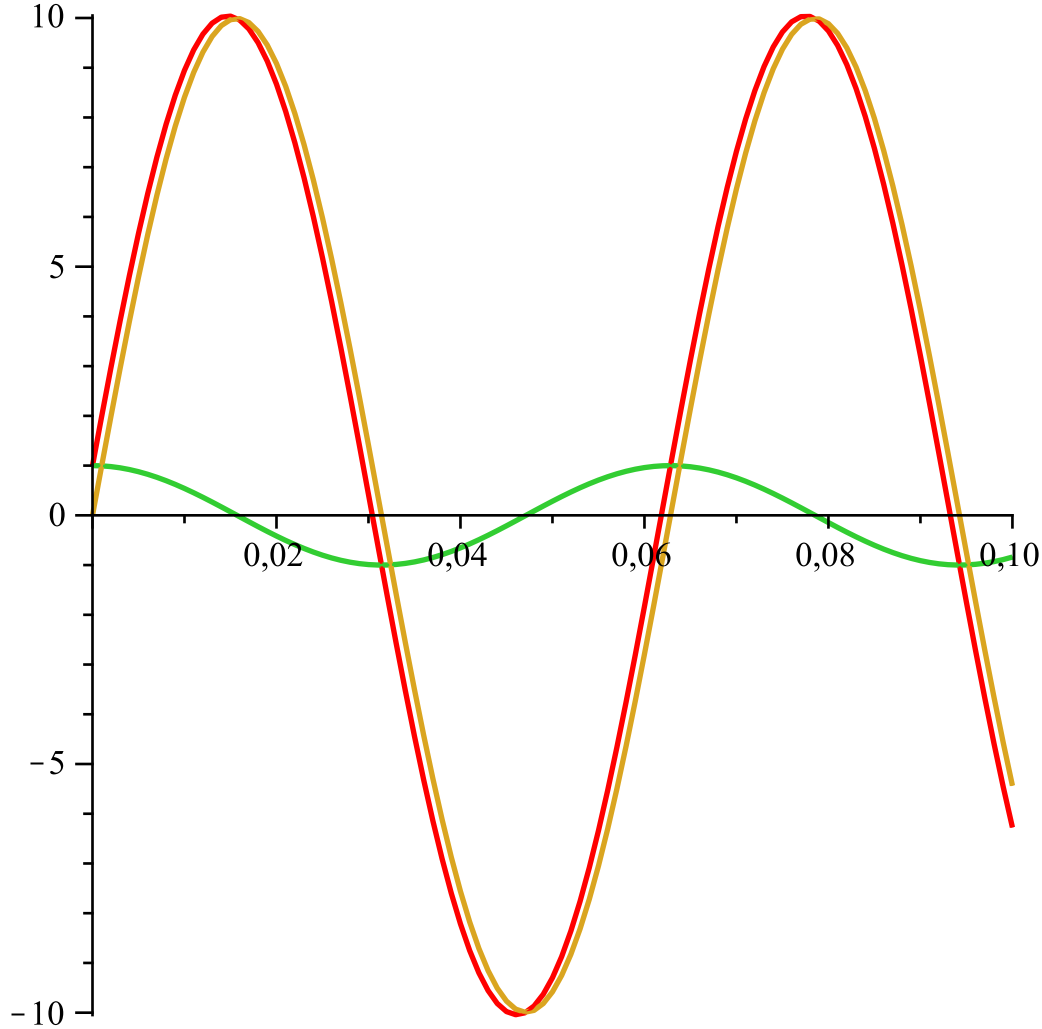
$$ALG := \left\{ I2\_N = \frac{u\_N}{R},\ I1\_N = \frac{u\_N + I3\_N\,R}{R} \right\} \qquad (1.3)$$

```
> CN := solve( subs(ALG,subs( diff(I3(t),t)=(I3_N-I3_O)/DT,
                I2(t)=(I2_N+I2_O)/2,
                u(t)=(u_N+u_O)/2, EQ1 )), {I3_N} ) ;
```

(1.4)

$$CN := \left\{ I3\_N = \frac{1}{2} \frac{2\, L\, I3\_O + u\_N\, DT + u\_O\, DT}{L} \right\} \tag{1.4}$$

```
> AdvanceCN := proc( I3_init, dt, N )
    local kk, SUBS, I1, I2, I3, a, b, c ;
    SUBS := { R = 1, L = 0.001, u_N=cos(100*0), I3_N=I3_init }
  ;
    I1 := subs( SUBS, [[0,subs(ALG,I1_N)]] );
    I2 := subs( SUBS, [[0,subs(ALG,I2_N)]] );
    I3 := [[0,I3_init]] ;
    for kk from 1 to N do
      SUBS := { R = 1,
                L = 0.001,
                DT=dt,
                I1_O=I1[-1][2],
                I2_O=I2[-1][2],
                I3_O=I3[-1][2],
                u_O =cos(100*(kk-1)*dt),
                u_N =cos(100*kk*dt)} ;
      a  := evalf(subs(SUBS,subs(CN,I3_N))) ;
      b  := evalf(subs(I3_N=a,subs(SUBS,subs(ALG,I1_N)))) ;
      c  := evalf(subs(I3_N=a,subs(SUBS,subs(ALG,I2_N)))) ;
      I1 := [op(I1),[kk*dt,b]];
      I2 := [op(I2),[kk*dt,c]];
      I3 := [op(I3),[kk*dt,a]];
    end ;
    [I1,I2,I3];
  end proc:
> RES := AdvanceCN( 0, 0.001, 100 ):
> CNPLOT := plot( RES, thickness=[2,2,2] ):
> display( CNPLOT, ESATTA );
```



Errror computation and order estimation

```
> N1 := 100 ; h1 := 0.1 / N1 ;
  N2 := 200 ; h2 := 0.1 / N2 ;
  N3 := 400 ; h3 := 0.1 / N3 ;
  N4 := 800 ; h4 := 0.1 / N4 ;
```

$$N1 := 100$$

$$h1 := 0.001000000000$$

$$N2 := 200$$

$$h2 := 0.0005000000000$$

$$N3 := 400$$

$$h3 := 0.0002500000000$$

$$N4 := 800$$

$$h4 := 0.0001250000000 \quad \textbf{(1.5)}$$

```
> RES1 := AdvanceCN( 0, h1, N1):
  RES2 := AdvanceCN( 0, h2, N2):
  RES3 := AdvanceCN( 0, h3, N3):
  RES4 := AdvanceCN( 0, h4, N4):
> E1 := max(seq( abs(I3_esatta(RES1[1][k][1])-RES1[3][k][2]),
  k=1..N1)) ;
```

```
E2 := max(seq( abs(I3_esatta(RES2[1][k][1])-RES2[3][k][2]),
k=1..N2)) ;
E3 := max(seq( abs(I3_esatta(RES3[1][k][1])-RES3[3][k][2]),
k=1..N3)) ;
E4 := max(seq( abs(I3_esatta(RES4[1][k][1])-RES4[3][k][2]),
k=1..N4)) ;
```

$$E1 := 0.008334086$$

$$E2 := 0.002083463$$

$$E3 := 0.000520821$$

$$E4 := 0.000130278 \qquad (1.6)$$

```
> evalf(log(E1/E2)/log(2)) ;
  evalf(log(E2/E3)/log(2)) ;
  evalf(log(E3/E4)/log(2)) ;
```

$$2.000040508$$

$$2.000123953$$

$$1.999194144 \qquad (1.7)$$

Crank-Nicolson without coordinate partitioning

Solve numerically with Crank-Nicolson

```
> SUBSCN := { u(t)  = (u_N+u_O)/2,
              I1(t) = (I1_O+I1_N)/2,
              I2(t) = (I2_O+I2_N)/2,
              I3(t) = (I3_O+I3_N)/2 } ;
```

$$SUBSCN := \left\{u(t) = \frac{1}{2}\,u\_N + \frac{1}{2}\,u\_O, I1(t) = \frac{1}{2}\,I1\_O + \frac{1}{2}\,I1\_N, I2(t) = \frac{1}{2}\,I2\_O + \frac{1}{2}\,I2\_N, I3(t) = \frac{1}{2}\,I3\_O + \frac{1}{2}\,I3\_N\right\} \qquad (2.1)$$

```
> CN1 := subs(SUBSCN, subs( diff(I3(t),t)=(I3_N-I3_O)/DT,EQ1));
  CN2 := subs(SUBSCN,EQ2);
  CN3 := subs(SUBSCN,EQ3);
```

$$CN1 := \frac{L\,(I3\_N - I3\_O)}{DT} - \frac{1}{2}\,u\_N - \frac{1}{2}\,u\_O$$

$$CN2 := \frac{1}{2}\,I1\_O + \frac{1}{2}\,I1\_N - \frac{1}{2}\,I2\_O - \frac{1}{2}\,I2\_N - \frac{1}{2}\,I3\_O - \frac{1}{2}\,I3\_N$$

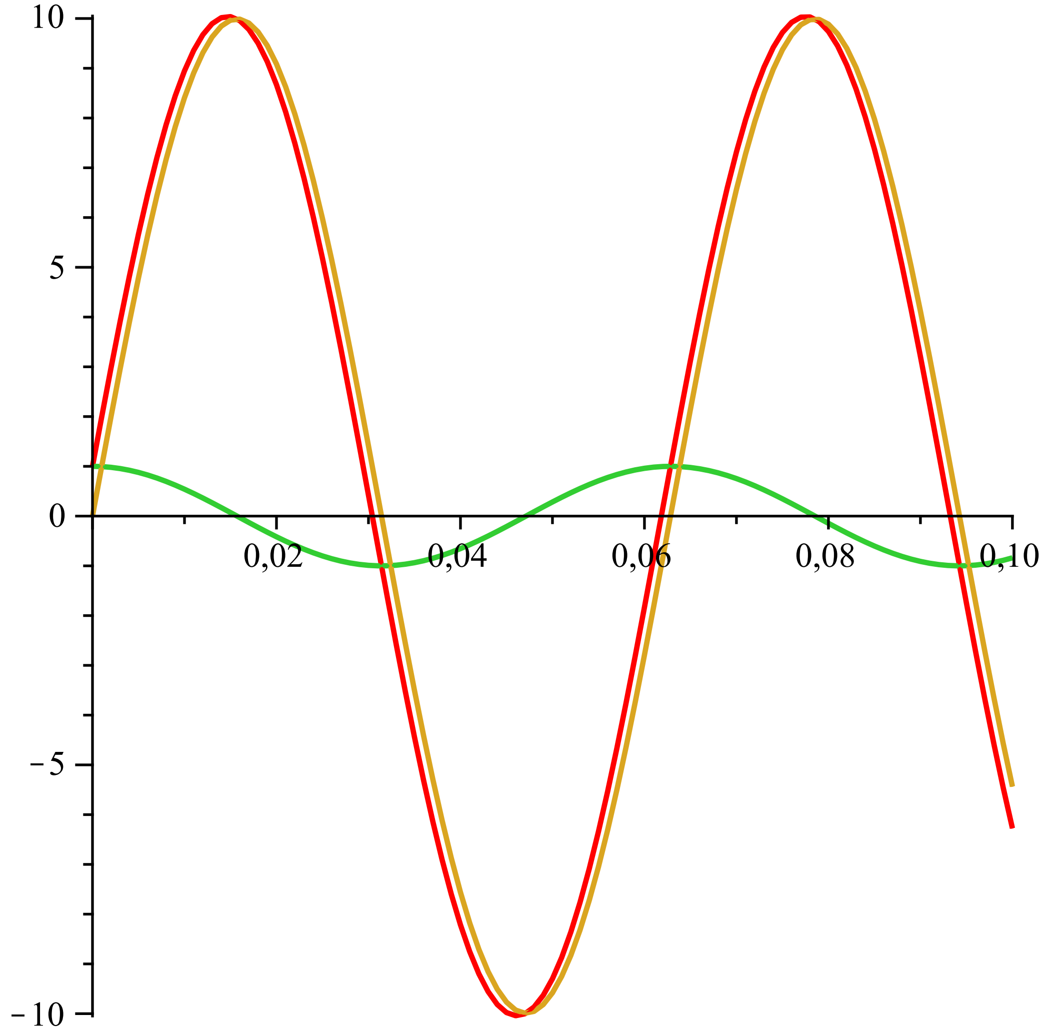
$$CN3 := \frac{1}{2}\,u\_N + \frac{1}{2}\,u\_O - R\left(\frac{1}{2}\,I2\_O + \frac{1}{2}\,I2\_N\right) \qquad (2.2)$$

```
> CNDAE := solve( {CN1,CN2,CN3}, {I1_N,I2_N,I3_N} ) ;
```

$$CNDAE := \left\{I3\_N = \frac{1}{2}\,\frac{2\,L\,I3\_O + u\_N\,DT + u\_O\,DT}{L}, I2\_N = -\frac{-u\_N - u\_O + R\,I2\_O}{R}, I1\_N = \frac{1}{2}\,\frac{-2\,I1\_O\,R\,L + 2\,u\_N\,L + 2\,u\_O\,L + 4\,I3\_O\,R\,L + R\,u\_N\,DT + R\,u\_O\,DT}{L\,R}\right\} \qquad (2.3)$$

```
> AdvanceCNDAE := proc( I1_init, I2_init, I3_init, dt, N )
    local kk, SUBS, I1, I2, I3, a, b, c ;
    SUBS := { R = 1, L = 0.001 } ;
```

```
    I1 := [[0,I1_init]] ;
    I2 := [[0,I2_init]] ;
    I3 := [[0,I3_init]] ;
    for kk from 1 to N do
      SUBS := { R    = 1,
                L    = 0.001,
                DT   = dt,
                I1_O = I1[-1][2],
                I2_O = I2[-1][2],
                I3_O = I3[-1][2],
                u_O  = cos(100*(kk-1)*dt),
                u_N  = cos(100*kk*dt)} ;
      a  := evalf(subs(SUBS,subs(CNDAE,I1_N))) ;
      b  := evalf(subs(SUBS,subs(CNDAE,I2_N))) ;
      c  := evalf(subs(SUBS,subs(CNDAE,I3_N))) ;
      I1 := [op(I1),[kk*dt,a]];
      I2 := [op(I2),[kk*dt,b]];
      I3 := [op(I3),[kk*dt,c]];
    end ;
    [I1,I2,I3];
  end proc:
> RES := AdvanceCNDAE( 1, 1, 0.0001, 0.001, 100 ):
> CNPPLTDAE := plot( RES, thickness=[2,2,2] ):
> display( CNPPLTDAE, ESATTA );
```



Errror computation and order estimation

```
> N1 := 100 ; h1 := 0.1 / N1 ;
  N2 := 200 ; h2 := 0.1 / N2 ;
  N3 := 400 ; h3 := 0.1 / N3 ;
  N4 := 800 ; h4 := 0.1 / N4 ;
```

$$N1 := 100$$

$$h1 := 0.001000000000$$

$$N2 := 200$$

$$h2 := 0.0005000000000$$

$$N3 := 400$$

$$h3 := 0.0002500000000$$

$$N4 := 800$$

$$h4 := 0.0001250000000 \quad (2.4)$$

```
> RES1 := AdvanceCNDAE( 1, 1, 0, h1, N1):
  RES2 := AdvanceCNDAE( 1, 1, 0, h2, N2):
  RES3 := AdvanceCNDAE( 1, 1, 0, h3, N3):
  RES4 := AdvanceCNDAE( 1, 1, 0, h4, N4):
```

```
> E1 := max(seq( abs(I3_esatta(RES1[1][k][1])-RES1[3][k][2]),
  k=1..N1)) ;
  E2 := max(seq( abs(I3_esatta(RES2[1][k][1])-RES2[3][k][2]),
  k=1..N2)) ;
  E3 := max(seq( abs(I3_esatta(RES3[1][k][1])-RES3[3][k][2]),
  k=1..N3)) ;
  E4 := max(seq( abs(I3_esatta(RES4[1][k][1])-RES4[3][k][2]),
  k=1..N4)) ;
```

$$E1 := 0.008334086$$

$$E2 := 0.002083463$$

$$E3 := 0.000520821$$

$$E4 := 0.000130278 \quad (2.5)$$

```
> evalf(log(E1/E2)/log(2)) ;
  evalf(log(E2/E3)/log(2)) ;
  evalf(log(E3/E4)/log(2)) ;
```

$$2.000040508$$

$$2.000123953$$

$$1.999194144 \quad (2.6)$$