Soluzioni del compito di

Metodi Matematici e Calcolo per Ingegneria

del 19 giugno 2006

Enrico Bertolazzi

Trasformata di Laplace

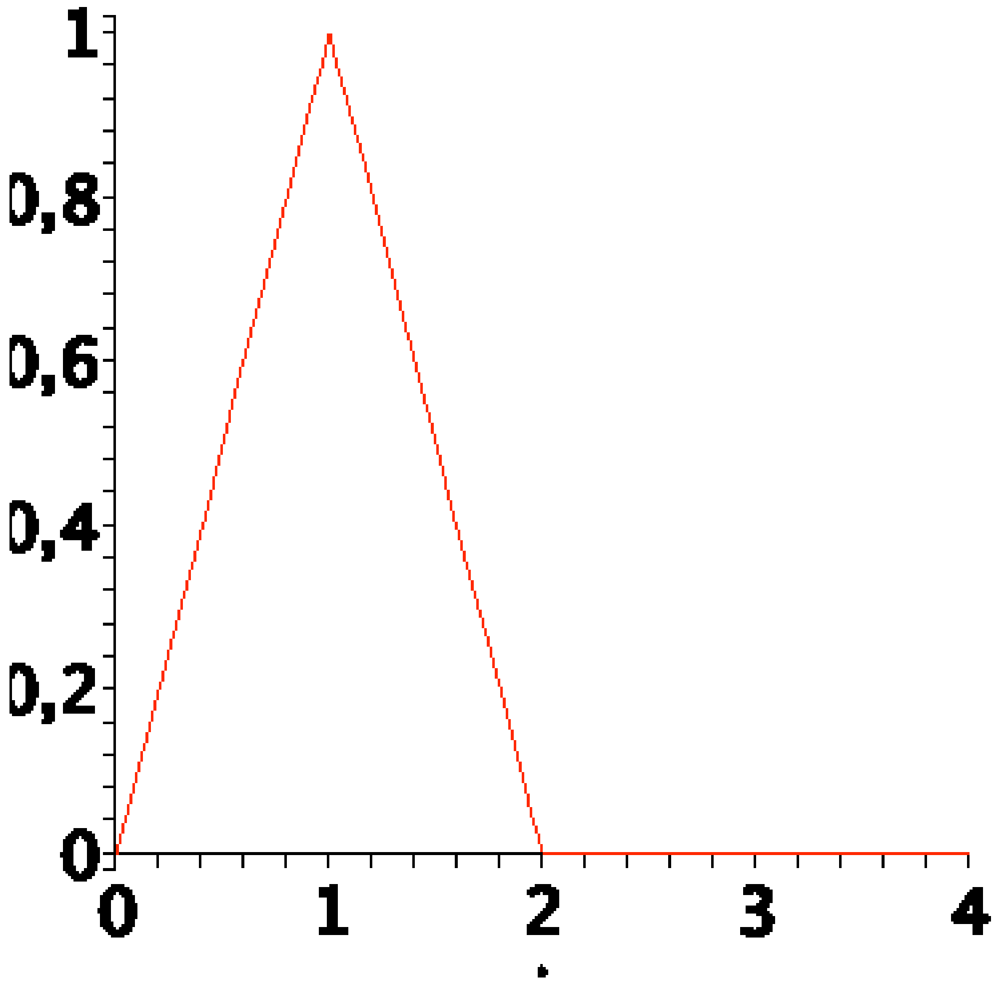
```
> restart:
  with(inttrans) :
```

Data la seguente funzione

```
> f := t -> t - 2*(t-1)*Heaviside(t-1)+(t-2)*Heaviside(t-2);
```

$$f := t \rightarrow t - 2\,(t-1)\,\mathrm{Heaviside}(t-1) + (t-2)\,\mathrm{Heaviside}(t-2)$$

```
> plot(f(t),t=0..4,axesfont=[helvetica,24]);
```



Usando le regole di trasformazione calcolare le trasformate delle funzioni

```
> f1 := t -> f(t/3) ;
  f2 := t -> f(t/2)*exp(-2*t) ;
  f3 := unapply( diff(f(t),t), t) ;
```

$$f1 := t \to f\left(\frac{1}{3}t\right)$$

$$f2 := t \to f\left(\frac{1}{2}t\right)e^{(-2t)}$$

$$f3 := t \to 1 - 2\,\mathrm{Heaviside}(t-1) - 2\,(t-1)\,\mathrm{Dirac}(t-1) + \mathrm{Heaviside}(t-2) + (t-2)\,\mathrm{Dirac}(t-2)$$

Trasformate con le primitive Maple

```
> F := unapply( laplace(f(t),t,s), s );
```

$$F := s \to \frac{1 - 2\,e^{(-s)} + e^{(-2s)}}{s^2}$$

Calcolo le altre trasformate con le regole di trasformazione nelle tabelle

f(t/3) è un cambio di scala t --> t/3 diventa 3*F(3s)

```
> 3*F(3*s);
```

$$\frac{1 - 2\,e^{(-3\,s)} + e^{(-6\,s)}}{3\,s^2}$$

Controllo con le primitive di Maple

```
> laplace(f1(t),t,s);
```

$$\frac{1 - 2\,e^{(-3\,s)} + e^{(-6\,s)}}{3\,s^2}$$

f(t/2)*exp(-2*t) è un cambio di scala t --> t/2 per un esponenziale.
Spezzo la regola di trasformazione in vari passaggi:

- dalla regola del cambio di scala ho L(f(t/2)) = 2*F(2*s)
- definisco g(t) := f(t/2)
- definisco G(s)=2*F(2*s) la trasformata di g(t)
- la trasformata di g(t)*exp(-2*t) diventa G(s+2)
- quindi L(f(t/2)*exp(-2*t))=L(g(t)*exp(-2*t))=G(s+2)=2*F(2*(s+2))

```
> simplify(2*F(2*(s+2)));
```

$$\frac{1 - 2\,e^{(-2\,s-4)} + e^{(-4\,s-8)}}{2\,(s+2)^2}$$

Controllo con le primitive di Maple

```
> laplace(f2(t),t,s);
```

$$\frac{1 - 2\,e^{(-2\,s-4)} + e^{(-4\,s-8)}}{2\,(s+2)^2}$$

f '(t) è una derivazione.
Applico la regola di trasformazione: L(f '(t)) := s*F(s)-f(0)

```
> s*F(s)-f(0);
```

$$\frac{1 - 2\,e^{(-s)} + e^{(-2\,s)}}{s}$$

Controllo con le primitive di Maple

```
> laplace(f3(t),t,s);
```

$$\frac{1 - 2\,e^{(-s)} + e^{(-2\,s)}}{s}$$

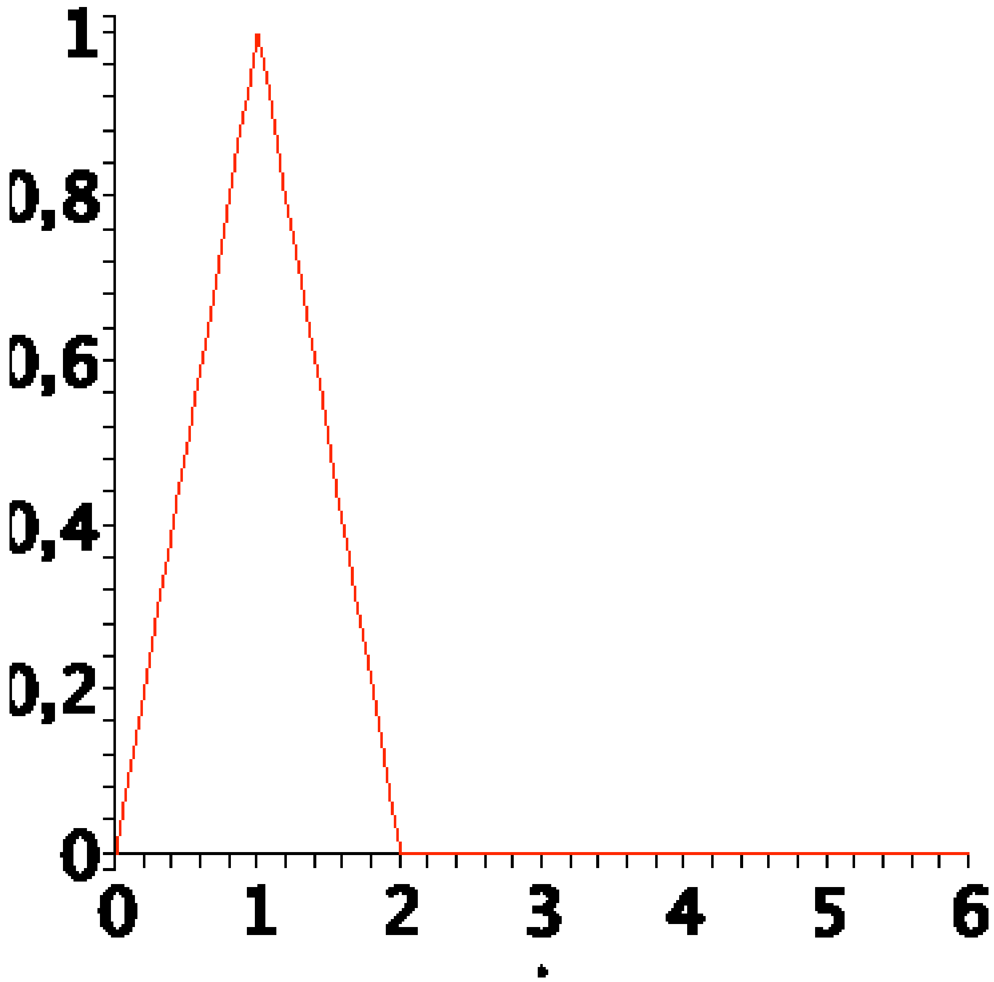
Soluzione di ODE con Laplace

```
> restart:
  with(inttrans) :
```

```
> src := t -> t - 2*(t-1)*Heaviside(t-1)+(t-2)*Heaviside(t-2);
```

$$src := t \rightarrow t - 2\,(t-1)\,\mathrm{Heaviside}(t-1) + (t-2)\,\mathrm{Heaviside}(t-2)$$

```
> plot(src(t),t=0..6,axesfont=[helvetica,24]);
```



Data la seguente equazione differenziale

```
> ode := diff(y(x),x)=src(x) ;
```

$$ode := \frac{d}{dx}\, y(x) = x - 2\,(x - 1)\ \mathrm{Heaviside}(x - 1) + (x - 2)\ \mathrm{Heaviside}(x - 2)$$

Con dato iniziale

```
> y0 := 2 ;
```

$$y0 := 2$$

Calcolare le soluzioni con le trasformate di Laplace.

Trasformo la equazione differenziale con la trasformata di Laplace

```
> sode := laplace(ode,x,s) ;
```

$$sode := s\ laplace(y(x), x, s) - y(0) = \frac{1 - 2\,\mathbf{e}^{(-s)} + \mathbf{e}^{(-2\,s)}}{s^2}$$

```
> subs(laplace(y(x),x,s)=y(s),sode) ;
```

$$s\ y(s) - y(0) = \frac{1 - 2\,\mathbf{e}^{(-s)} + \mathbf{e}^{(-2\,s)}}{s^2}$$

Risolvo la equazione per y(s)

```
> lode := isolate(sode,laplace(y(x),x,s));
```

$$lode := laplace(y(x), x, s) = \frac{\frac{1 - 2\,e^{(-s)} + e^{(-2\,s)}}{s^2} + y(0)}{s}$$

Applico le condizioni iniziali ottenendo y(s)

```
> ly := expand(subs( y(0)=y0,rhs(lode)) ;
```

$$ly := \frac{1}{s^3} - \frac{2}{s^3\,e^s} + \frac{1}{s^3\,(e^s)^2} + \frac{2}{s}$$

Espansione in fratti semplici

```
> convert(ly, fullparfrac, s);
```

$$\frac{1}{s^3} - \frac{2}{s^3\,e^s} + \frac{1}{s^3\,(e^s)^2} + \frac{2}{s}$$

Antitrasformo per ottenere la equazione y(x)

```
> res := invlaplace(ly,s,t) ;
```

$$res := \frac{1}{2}\,t^2 - \mathrm{Heaviside}(t-1)\,(t-1)^2 + \frac{1}{2}\,\mathrm{Heaviside}(t-2)\,(t-2)^2 + 2$$

Definendo pos(x) = x se x>0 pos(x)=0 se x <= 0 abbiamo

```
> subs(Heaviside(t-1)*(t-1)^2=pos(t-1)^2,Heaviside(t-2)*(t-2)^2=pos(t-2
)^2,res);
```

$$\frac{1}{2}\,t^2 - pos(t-1)^2 + \frac{1}{2}\,pos(t-2)^2 + 2$$

Soluzione di un sistema di ODE con Laplace

```
> restart:
  with(inttrans) :
```

Dato il seguente sistema di equazioni differenziali

```
> yp, zp, wp := diff(y(t),t),diff(z(t),t),diff(w(t),t);
```

$$yp, zp, wp := \frac{d}{dt}\,y(t),\ \frac{d}{dt}\,z(t),\ \frac{d}{dt}\,w(t)$$

```
> ode1 := yp    -zp       = t -
  2*(t-1)*Heaviside(t-1)+(t-2)*Heaviside(t-2) ;
  ode2 := -yp +2*zp -wp   = 0 ;
  ode3 :=     -zp   +2*wp = 0 ;
```

$$ode1 := \left(\frac{d}{dt}\,y(t)\right) - \left(\frac{d}{dt}\,z(t)\right) = t - 2\,(t-1)\,\mathrm{Heaviside}(t-1) + (t-2)\,\mathrm{Heaviside}(t-2)$$

$$ode2 := -\left(\frac{d}{dt}\,y(t)\right) + 2\left(\frac{d}{dt}\,z(t)\right) - \left(\frac{d}{dt}\,w(t)\right) = 0$$

$$ode3 := -\left(\frac{d}{dt} z(t)\right) + 2\left(\frac{d}{dt} w(t)\right) = 0$$

Con dato iniziale

```
> y0, z0, w0 := 1, 2, 1 ;
```

$$y0, z0, w0 := 1, 2, 1$$

Calcolare le soluzioni con le trasformate di Laplace.

Trasformo le equazioni differenziale con la trasformata di Laplace

```
> sode1 := laplace(ode1,t,s) ;
  sode2 := laplace(ode2,t,s) ;
  sode3 := laplace(ode3,t,s) ;
```

$$sode1 := s\ laplace(y(t), t, s) - y(0) - s\ laplace(z(t), t, s) + z(0) = \frac{1 - 2\,\mathbf{e}^{(-s)} + \mathbf{e}^{(-2\,s)}}{s^2}$$

$$sode2 := -s\ laplace(y(t), t, s) + y(0) + 2\,s\ laplace(z(t), t, s) - 2\,z(0) - s\ laplace(w(t), t, s) + w(0) = 0$$

$$sode3 := -s\ laplace(z(t), t, s) + z(0) + 2\,s\ laplace(w(t), t, s) - 2\,w(0) = 0$$

```
> subs(laplace(y(t),t,s)=y(s),
       laplace(z(t),t,s)=z(s),
       laplace(w(t),t,s)=w(s),
       sode1);
  subs(laplace(y(t),t,s)=y(s),
       laplace(z(t),t,s)=z(s),
       laplace(w(t),t,s)=w(s),
       sode2);
  subs(laplace(y(t),t,s)=y(s),
       laplace(z(t),t,s)=z(s),
       laplace(w(t),t,s)=w(s),
       sode3);
```

$$s\,y(s) - y(0) - s\,z(s) + z(0) = \frac{1 - 2\,\mathbf{e}^{(-s)} + \mathbf{e}^{(-2\,s)}}{s^2}$$

$$-s\,y(s) + y(0) + 2\,s\,z(s) - 2\,z(0) - s\,w(s) + w(0) = 0$$

$$-s\,z(s) + z(0) + 2\,s\,w(s) - 2\,w(0) = 0$$

Risolvo la equazione per y(s), z(s)

```
> ys,zs,ws := laplace(y(t),t,s),laplace(z(t),t,s),laplace(w(t),t,s);
```

$$ys, zs, ws := laplace(y(t), t, s), laplace(z(t), t, s), laplace(w(t), t, s)$$

```
> RES := solve({sode1,sode2,sode3},{ys,zs,ws});
```

$$RES := \left\{ laplace(w(t), t, s) = -\frac{-1 - \mathbf{e}^{(-2\,s)} + 2\,\mathbf{e}^{(-s)} - s^2\,w(0)}{s^3}, \right.$$

$$laplace(y(t), t, s) = \frac{3 + 3\,\mathbf{e}^{(-2\,s)} - 6\,\mathbf{e}^{(-s)} + y(0)\,s^2}{s^3},$$

$$laplace(z(t), t, s) = -\frac{-z(0)\, s^2 - 2 - 2\, \mathbf{e}^{(-2\, s)} + 4\, \mathbf{e}^{(-s)}}{s^3}$$

Applico le condizioni iniziali ottenendo y(s), z(s)

```
> SOL := subs(RES,y(0)=y0,z(0)=z0,w(0)=w0,<ys,zs,ws>);
```

$$SOL := \begin{bmatrix} \dfrac{3 + 3\, \mathbf{e}^{(-2\, s)} - 6\, \mathbf{e}^{(-s)} + s^2}{s^3} \\ -\dfrac{-2\, s^2 - 2 - 2\, \mathbf{e}^{(-2\, s)} + 4\, \mathbf{e}^{(-s)}}{s^3} \\ -\dfrac{-1 - \mathbf{e}^{(-2\, s)} + 2\, \mathbf{e}^{(-s)} - s^2}{s^3} \end{bmatrix}$$

Antitrasformo per ottenere y(x), z(x)

```
> yy := invlaplace(SOL[1],s,x) ;
  zz := invlaplace(SOL[2],s,x) ;
  ww := invlaplace(SOL[3],s,x) ;
```

$$yy := \frac{3}{2} x^2 + \frac{3}{2} \mathrm{Heaviside}(x - 2)\, (x - 2)^2 - 3\, \mathrm{Heaviside}(x - 1)\, (x - 1)^2 + 1$$

$$zz := 2 + x^2 + \mathrm{Heaviside}(x - 2)\, (x - 2)^2 - 2\, \mathrm{Heaviside}(x - 1)\, (x - 1)^2$$

$$ww := \frac{1}{2} x^2 + \frac{1}{2} \mathrm{Heaviside}(x - 2)\, (x - 2)^2 - \mathrm{Heaviside}(x - 1)\, (x - 1)^2 + 1$$

Definendo pos(x) = x se x>0 pos(x)=0 se x <= 0 abbiamo

```
> subs(Heaviside(x-1)*(x-1)^2=pos(x-1)^2,Heaviside(x-2)*(x-2)^2=pos(t-2
  )^2,
       <yy,zz,ww>);
```

$$\begin{bmatrix} \frac{3}{2} x^2 + \frac{3}{2} pos(t - 2)^2 - 3\, pos(x - 1)^2 + 1 \\ 2 + x^2 + pos(t - 2)^2 - 2\, pos(x - 1)^2 \\ \frac{1}{2} x^2 + \frac{1}{2} pos(t - 2)^2 - pos(x - 1)^2 + 1 \end{bmatrix}$$

Soluzione di ricorrenza con trasformata zeta

```
> restart:
```

Risolvere la seguente ricorrenza

```
> RIC := f(n+2) = 4*f(n+1) - 4*f(n) - n*(n-1) ;
```

$$RIC := f(n + 2) = 4\, f(n + 1) - 4\, f(n) - n\, (n - 1)$$

Con dato iniziale

```
> INI := f(0)=0,f(1)=2;
```

$$INI := f(0) = 0, f(1) = 2$$

Usando le primitive di maple:

```
> rsolve({RIC,INI}, f(k));
```

$$(k+1)\,2^k + 6\,2^k + (-k-1)\,2^k - 2\,(k+1)\left(\frac{1}{2}k+1\right) - 4$$

```
> simplify(%);
```

$$6\,2^k - k^2 - 3\,k - 6$$

Usando la Z-trasformata

```
> zRIC := ztrans(RIC,n,z);
```

$$zRIC := z^2\,ztrans(f(n), n, z) - f(0)\,z^2 - f(1)\,z = 4\,z\,ztrans(f(n), n, z) - 4\,f(0)\,z - 4\,ztrans(f(n), n, z) - \frac{z\,(z+1)}{(z-1)^3} + \frac{z}{(z-1)^2}$$

Ricavo f(z)

```
> zRICrhs := isolate(zRIC,ztrans(f(n),n,z));
```

$$zRICrhs := ztrans(f(n), n, z) = \frac{f(0)\,z^2 + f(1)\,z - 4\,f(0)\,z - \frac{z\,(z+1)}{(z-1)^3} + \frac{z}{(z-1)^2}}{z^2 - 4\,z + 4}$$

Applico le condizioni iniziali

```
> zRICrhsINI := subs(INI,zRICrhs);
```

$$zRICrhsINI := ztrans(f(n), n, z) = \frac{2\,z - \frac{z\,(z+1)}{(z-1)^3} + \frac{z}{(z-1)^2}}{z^2 - 4\,z + 4}$$

Conversione in fratti semplici

```
> convert(%, parfrac);
```

$$ztrans(f(n), n, z) = -\frac{10}{z-1} - \frac{2}{(z-1)^3} + \frac{12}{z-2} - \frac{6}{(z-1)^2}$$

Inversione della Z-trasformata

```
> invztrans(zRICrhsINI,z,k) ;
```

$$f(k) = -6 - 3\,k - k^2 + 6\,2^k$$

Soluzione di un sistema non lineare con Newton

```
> restart:
  with(VectorCalculus):
Warning, the assigned names `<,>` and `<|>` now have a global binding

Warning, these protected names have been redefined and unprotected:
```

```
`*`, `+`, `.`, D, Vector, diff, int, limit, series
```

Sistema non lineare

```
> f := 2*x-y + x*y + 1 ;
  g :=     x + 2*y - x*y - 2 ;
```

$$f := 2\,x - y + x\,y + 1$$

$$g := x + 2\,y - x\,y - 2$$

Soluzione esatta

```
> solve({f,g},{x,y}) ;
```

$$\{y = 1, x = 0\}, \{x = 2, y = -5\}$$

Matrice Jacobiano

```
> J := Jacobian([f,g],[x,y]) ;
```

$$J := \begin{bmatrix} 2 + y & -1 + x \\ 1 - y & -x + 2 \end{bmatrix}$$

Schema di Newton

```
> Newton_update := simplify(<x,y>-J^(-1).<f,g>);
```

$$Newton\_update := -\frac{x\,(-1 + y)}{3\,x - 5 - y}\,e_x + \frac{3\,x\,y - y - 5}{3\,x - 5 - y}\,e_y$$

Schema di Newton per questo sistema non lineare

```
> x[k+1]=simplify(subs(x=x[k],y=y[k],Newton_update[1])) ;
  y[k+1]=simplify(subs(x=x[k],y=y[k],Newton_update[2])) ;
```

$$x_{k+1} = -\frac{x_k\,(-1 + y_k)}{3\,x_k - 5 - y_k}$$

$$y_{k+1} = \frac{3\,x_k\,y_k - y_k - 5}{3\,x_k - 5 - y_k}$$

Tre iterate a partire da (1,2)

```
> x[0],y[0]:= 1,2 ;
```

$$x_0, y_0 := 1, 2$$

Prima iterata

```
> x[1] := evalf(subs(x=x[0],y=y[0],Newton_update[1])) ;
  y[1] := evalf(subs(x=x[0],y=y[0],Newton_update[2])) ;
```

$$x_1 := 0.2500000000$$

$$y_1 := 0.2500000000$$

Seconda iterata

```
> x[2] := evalf(subs(x=x[1],y=y[1],Newton_update[1])) ;
  y[2] := evalf(subs(x=x[1],y=y[1],Newton_update[2])) ;
```

$$x_2 := -0.04166666668$$

$$y_2 := 1.125000000$$

Terza iterata

```
> x[3] := evalf(subs(x=x[2],y=y[2],Newton_update[1])) ;
  y[3] := evalf(subs(x=x[2],y=y[2],Newton_update[2])) ;
```

$$x_3 := -0.0008333333336$$

$$y_3 := 1.002500000$$

Problema di Minimo Vincolato

```
> restart:
  with(LinearAlgebra):
  with(Optimization):
  with(VectorCalculus):
Warning, the names `&x`, CrossProduct and DotProduct have been rebound

Warning, the assigned names `<,>` and `<|>` now have a global binding

Warning, these protected names have been redefined and unprotected:
`*`, `+`, `.`, D, Vector, diff, int, limit, series
```

Minimizzare la seguente funzione

```
> f := (x-y)^2+(x-z)^2+(y-z)^2;
```

$$f := (x - y)^2 + (x - z)^2 + (y - z)^2$$

Soggetta ai vicoli

```
> v := [(x+y+z)*y=1,x-z=1] ;
```

$$v := [(x + y + z)\, y = 1, x - z = 1]$$

Soluzione con le primitive Maple

```
> Minimize(f, v );
```

$$[1.49999999999999956, [x = 1.07735026918962573, z = 0.0773502691896258560, y = 0.577350269189625731]]$$

Uso dei moltiplicatori di Lagrange

```
> v1 := lhs(v[1])-rhs(v[1]) ;
  v2 := lhs(v[2])-rhs(v[2]) ;
```

$$v1 := (x + y + z)\, y - 1$$

$$v2 := x - z - 1$$

```
> g := f - lambda*v1 - mu*v2 ;
```

$$g := (x - y)^2 + (x - z)^2 + (y - z)^2 - \lambda\, ((x + y + z)\, y - 1) - \mu\, (x - z - 1)$$

Sistema non lineare da risolvere

```
> F := Gradient(g,[x,y,z,lambda,mu]) ;
```

$$F := (4\, x - 2\, y - 2\, z - \lambda\, y - \mu)\, \overline{e}_x + (-2\, x + 4\, y - 2\, z - \lambda\, (2\, y + x + z))\, \overline{e}_y + (-2\, x + 4\, z - 2\, y - \lambda\, y + \mu)\, \overline{e}_z + (-(x + y + z)\, y + 1)\, \overline{e}_\lambda + (-x + z + 1)\, \overline{e}_\mu$$

Soluzioni del sistema non lineare

```
> _EnvExplicit := true ;
```

$$\_EnvExplicit := true$$

```
> RES :=
  op(sort([solve({F[1],F[2],F[3],F[4],F[5]},{x,y,z,lambda,mu})]));
```

$$RES := \left\{ y = \frac{1}{3}\sqrt{3},\, x = \frac{1}{2} + \frac{1}{3}\sqrt{3},\, \mu = 3,\, z = -\frac{1}{2} + \frac{1}{3}\sqrt{3},\, \lambda = 0 \right\},$$
$$\left\{ x = \frac{1}{2} + \frac{2}{3} I\sqrt{3},\, z = -\frac{1}{2} + \frac{2}{3} I\sqrt{3},\, y = -\frac{1}{3} I\sqrt{3},\, \lambda = -6,\, \mu = 3 \right\},$$
$$\left\{ x = \frac{1}{2} - \frac{2}{3} I\sqrt{3},\, z = -\frac{1}{2} - \frac{2}{3} I\sqrt{3},\, \lambda = -6,\, \mu = 3,\, y = \frac{1}{3} I\sqrt{3} \right\},$$
$$\left\{ z = -\frac{1}{2} - \frac{1}{3}\sqrt{3},\, x = \frac{1}{2} - \frac{1}{3}\sqrt{3},\, y = -\frac{1}{3}\sqrt{3},\, \mu = 3,\, \lambda = 0 \right\}$$

Prima soluzione

```
> S1 := RES[1];
```

$$S1 := \left\{ y = \frac{1}{3}\sqrt{3},\, x = \frac{1}{2} + \frac{1}{3}\sqrt{3},\, \mu = 3,\, z = -\frac{1}{2} + \frac{1}{3}\sqrt{3},\, \lambda = 0 \right\}$$

Seconda soluzione

```
> S2 := RES[4];
```

$$S2 := \left\{ z = -\frac{1}{2} - \frac{1}{3}\sqrt{3},\, x = \frac{1}{2} - \frac{1}{3}\sqrt{3},\, y = -\frac{1}{3}\sqrt{3},\, \mu = 3,\, \lambda = 0 \right\}$$

Controllo proprietà di minimo

```
> Hf  := Hessian(f,[x,y,z]):
  Hv1 := Hessian(v1,[x,y,z]):
  Hv2 := Hessian(v2,[x,y,z]):
  Hf, Hv1, Hv2 ;
```

$$\begin{bmatrix} 4 & -2 & -2 \\ -2 & 4 & -2 \\ -2 & -2 & 4 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

```
> JH := Jacobian([v1,v2],[x,y,z]) ;
  NH := NullSpace(JH) ;
```

$$JH := \begin{bmatrix} y & 2y + x + z & y \\ 1 & 0 & -1 \end{bmatrix}$$

$$NH := \left\{ \begin{bmatrix} 1 \\ -\dfrac{2y}{2y + x + z} \\ 1 \end{bmatrix} \right\}$$

Controllo minimo/massimo locale primo punto

```
> lambda1 := subs(S1,lambda);
  mu1     := subs(S1,mu);
```

$$\lambda 1 := 0$$

$$\mu 1 := 3$$

Calcolo l'Hessiano nel punto stazionario

```
> Hf1 := simplify(subs(S1,Hf - lambda1. Hv1 - mu1. Hv2)) ;
```

$$Hf1 := \begin{bmatrix} 4 & -2 & -2 \\ -2 & 4 & -2 \\ -2 & -2 & 4 \end{bmatrix}$$

L'Hessiano è semidefinto positivo!, devo controllare nello spazio dei vincoli

```
> evalf(Eigenvalues(Hf1));
```

$$\begin{bmatrix} 0. \\ 6. \\ 6. \end{bmatrix}$$

Cerco nello spazio dei vincoli:

```
> Z1 := subs(S1,op(NH)) ;
```

$$Z1 := \begin{bmatrix} 1 \\ \frac{-1}{2} \\ 1 \end{bmatrix}$$

E' positivo per ogni alpha, quindi è un minimo locale

```
> simplify(Transpose(alpha.Z1).Hf1.(alpha.Z1)) ;
```

$$9\,\alpha^2$$

```
> subs(S1,f);
```

$$\frac{3}{2}$$

Controllo minimo/massimo locale secondo punto

```
> lambda2 := subs(S2,lambda);
  mu2     := subs(S2,mu);
```

$$\lambda 2 := 0$$

$$\mu 2 := 3$$

```
> Hf2 := simplify(subs(S2,Hf - lambda2. Hv1 - mu2. Hv2)) ;
```

$$Hf2 := \begin{bmatrix} 4 & -2 & -2 \\ -2 & 4 & -2 \\ -2 & -2 & 4 \end{bmatrix}$$

L'Hessiano è semidefinito positivo!, devo controllare nello spazio dei vincoli

```
> evalf(Eigenvalues(Hf2));
```

$$\begin{bmatrix} 0. \\ 6. \\ 6. \end{bmatrix}$$

Cerco nello spazio dei vincoli:

```
> Z2 := subs(S2,op(NH)) ;
```

$$Z2 := \begin{bmatrix} 1 \\ \frac{-1}{2} \\ 1 \end{bmatrix}$$

E' positivo per ogni alpha, quindi è un minimo locale

```
> simplify(Transpose(alpha.Z2).Hf2.(alpha.Z2)) ;
```

$$9\,\alpha^2$$

```
> subs(S2,f);
```

$$\frac{3}{2}$$

Approssimazione di un problema del calcolo delle variazioni

```
> restart:
```

Integrale da minimizzare

```
> int(y(x)/(1+diff(y(x),x)),x=0..1);
```

$$\int_0^1 \frac{y(x)}{1+\left(\frac{d}{dx}\,y(x)\right)}\,dx$$

Condizioni al contorno

```
> ya,yb := 1,1 ;
```

$$ya, yb := 1, 1$$

```
> n := 4 ;
```

$$n := 4$$

```
> h := 1/n ;
```

$$h := \frac{1}{4}$$

```
> F := sum( (y[k+1]+y[k])/2*(1+((y[k+1]-y[k])/h)^2),k=0..n-1);
```

$$F := \frac{1}{2}(y_1+y_0)(1+16(y_1-y_0)^2)+\frac{1}{2}(y_2+y_1)(1+16(y_2-y_1)^2)+\frac{1}{2}(y_3+y_2)(1+16(y_3-y_2)^2)+\frac{1}{2}(y_4+y_3)(1+16(y_4-y_3)^2)$$

```
> eqns := Vector([seq(diff(F,y[k]),k=1..n-1),y[0]-ya,y[n]-yb]);
```

$$eqns := \begin{bmatrix} 1+8(y_1-y_0)^2+\frac{1}{2}(y_1+y_0)(32y_1-32y_0)+8(y_2-y_1)^2+\frac{1}{2}(y_2+y_1)(-32y_2+32y_1) \\ 1+8(y_2-y_1)^2+\frac{1}{2}(y_2+y_1)(32y_2-32y_1)+8(y_3-y_2)^2+\frac{1}{2}(y_3+y_2)(-32y_3+32y_2) \\ 1+8(y_3-y_2)^2+\frac{1}{2}(y_3+y_2)(32y_3-32y_2)+8(y_4-y_3)^2+\frac{1}{2}(y_4+y_3)(-32y_4+32y_3) \\ y_0-1 \\ y_4-1 \end{bmatrix}$$

```
> vars := [seq(y[k],k=0..n)];
```

$$vars := [y_0, y_1, y_2, y_3, y_4]$$

```
> with(VectorCalculus):with(LinearAlgebra):
Warning, the assigned names `<,>` and `<|>` now have a global binding

Warning, these protected names have been redefined and unprotected:
`*`, `+`, `.`, D, Vector, diff, int, limit, series
Warning, the names `&x`, CrossProduct and DotProduct have been rebound
```

```
> eqns_fun :=
  unapply(Vector([seq(simplify(subs(y[0]=ya,y[n]=yb,eqns[i]),sqrt,symbo
  lic),i=1..n-1)]),
  y[1],y[2],y[3]);
  vars_reduced := [seq(y[i],i=1..n-1)];
```

$eqns\_fun := (y\_1, y\_2, y\_3) \rightarrow rtable(1..3, \{(1) = -7+48\,y\_1^2-16\,y\_1-8\,y\_2^2-16\,y\_2\,y\_1,$
$(2) = 1+48\,y\_2^2-16\,y\_2\,y\_1-8\,y\_1^2-8\,y\_3^2-16\,y\_3\,y\_2,$
$(3) = -7+48\,y\_3^2-16\,y\_3\,y\_2-8\,y\_2^2-16\,y\_3\}, datatype = anything, subtype = Vector_{column},$
$storage = rectangular, order = Fortran\_order, attributes = [coords = cartesian])$

$$vars\_reduced := [y_1, y_2, y_3]$$

```
> J :=
  unapply(Matrix(simplify(Jacobian(eqns_fun(x,y,z),[x,y,z]),sqrt,symbol
  ic)),x,y,z) ;
```

$J := (x, y, z) \rightarrow rtable(1..3, 1..3, \{(2,1) = -16\,y-16\,x, (2,3) = -16\,z-16\,y, (2,2) = 96\,y-16\,x-16\,z$
$(1,1) = 96\,x-16-16\,y, (1,2) = -16\,y-16\,x, (3,2) = -16\,z-16\,y, (3,3) = 96\,z-16\,y-16\},$

datatype = *anything*, *subtype* = *Matrix*, *storage* = *rectangular*, *order* = *Fortran_order*)

```
> Newton_update := Z ->
  Z-LinearSolve(J(Z[1],Z[2],Z[3]),eqns_fun(Z[1],Z[2],Z[3]));
```

$Newton\_update := Z \to (VectorCalculus:-+)(Z, (VectorCalculus:-*)((LinearAlgebra:-LinearSolve)(J(Z_1, Z_2, Z_3), eqns\_fun(Z_1, Z_2, Z_3)), -1))$

```
> Z0 := <1,1,1>;
```

$$Z0 := e_x + e_y + e_z$$

```
> Z1 := evalf(Newton_update(Z0));
```

$$Z1 := 0.9531250000\, e_x + 0.9375000000\, e_y + 0.9531250000\, e_z$$

```
> Z2 := evalf(Newton_update(Z1));
```

$$Z2 := 0.951204019900000030\, e_x + 0.934561965500000035\, e_y + 0.951204019900000030\, e_z$$

```
> Z3 := evalf(Newton_update(Z2));
```

$$Z3 := 0.951200421400000051\, e_x + 0.934555355300000001\, e_y + 0.951200421400000051\, e_z$$

Verica del residuo di GRAD F

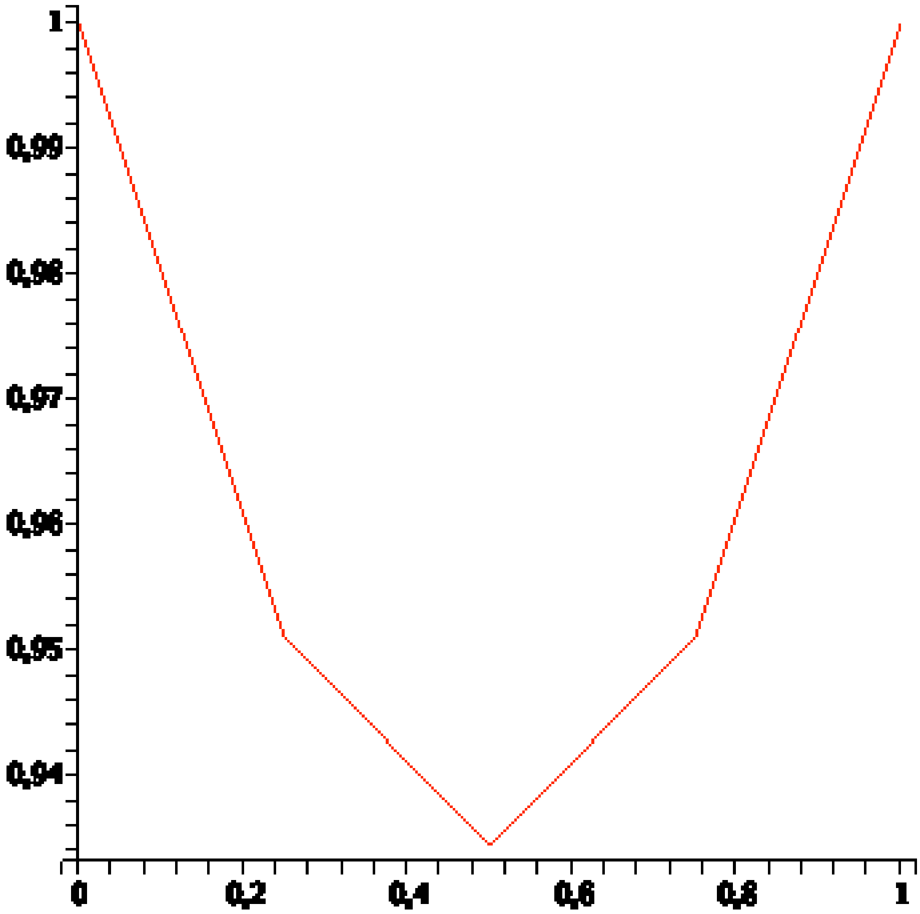
```
> subs(seq(y[i]=Z3[i],i=1..n-1),y[0]=ya,y[n]=yb,eqns) ;
```

$$\begin{bmatrix} -1.3\ 10^{-8} \\ -4.2\ 10^{-9} \\ -1.3\ 10^{-8} \\ 0 \\ 0 \end{bmatrix}$$

```
> yyy := [1,seq(Z3[k],k=1..3),1] ;
  xxx := [seq(k/n,k=0..n)]:
```

$yyy := [1, 0.951200421400000051, 0.934555355300000001, 0.951200421400000051, 1]$

```
> plot([seq([xxx[k],yyy[k]],k=1..nops(yyy))]);
```




```
> # Newton con molti piu punti!
> n := 100 ;
```

$$n := 100$$

```
> h := 1/n ;
```

$$h := \frac{1}{100}$$

```
> F := sum( (y[k+1]+y[k])/2*sqrt(1+((y[k+1]-y[k])/h)^2),k=0..n-1):
> eqns := [seq(diff(F,y[k]),k=1..n-1),y[0]-ya,y[n]-yb]:
> vars := [seq(y[k],k=0..n)];
```

$vars := [y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}, y_{12}, y_{13}, y_{14}, y_{15}, y_{16}, y_{17}, y_{18}, y_{19}, y_{20}, y_{21}, y_{22}, y_{23}, y_{24}, y_{25}, y_{26}, y_{27}, y_{28}, y_{29}, y_{30}, y_{31}, y_{32}, y_{33}, y_{34}, y_{35}, y_{36}, y_{37}, y_{38}, y_{39}, y_{40}, y_{41}, y_{42}, y_{43}, y_{44}, y_{45}, y_{46}, y_{47}, y_{48}, y_{49}, y_{50}, y_{51}, y_{52}, y_{53}, y_{54}, y_{55}, y_{56}, y_{57}, y_{58}, y_{59}, y_{60}, y_{61}, y_{62}, y_{63}, y_{64}, y_{65}, y_{66}, y_{67}, y_{68}, y_{69}, y_{70}, y_{71}, y_{72}, y_{73}, y_{74}, y_{75}, y_{76}, y_{77}, y_{78}, y_{79}, y_{80}, y_{81}, y_{82}, y_{83}, y_{84}, y_{85}, y_{86}, y_{87}, y_{88}, y_{89}, y_{90}, y_{91}, y_{92}, y_{93}, y_{94}, y_{95}, y_{96}, y_{97}, y_{98}, y_{99}, y_{100}]$

```
> eqns_fun :=
unapply(Vector([seq(simplify(subs(y[0]=ya,y[n]=yb,eqns[i]),sqrt,symbo
lic),i=1..n-1)]),
seq(y[k],k=1..n-1)):
```

```
vars_reduced := [seq(y[i],i=1..n-1)]:
```

```
> J :=
unapply(Matrix(simplify(Jacobian(eqns_fun(seq(y[k],k=1..n-1)),[seq(y[
k],k=1..n-1)]),sqrt,symbolic)),seq(y[k],k=1..n-1)):
```

```
> Newton_update := Z ->
Z-LinearSolve(J(seq(Z[k],k=1..n-1)),eqns_fun(seq(Z[k],k=1..n-1)));
```

$$Newton\_update := Z \rightarrow (VectorCalculus{:-}+)(Z, (VectorCalculus{:-}*)((LinearAlgebra{:-}LinearSolve)(J(seq(Z_k, k = 1 .. (VectorCalculus{:-}+)(n, (VectorCalculus{:-}*)(1, -1)))), eqns\_fun(seq(Z_k, k = 1 .. (VectorCalculus{:-}+)(n, (VectorCalculus{:-}*)(1, -1))))), -1))$$

```
> Z0 := <seq(1,k=1..n-1)>;
```

$$Z0 := \begin{bmatrix} \text{1 .. 99 Vector[column]} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

```
> Z1 := evalf(Newton_update(Z0));
```

$$Z1 := \begin{bmatrix} \text{1 .. 99 Vector[column]} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

```
> Z2 := evalf(Newton_update(Z1));
```

$$Z2 := \begin{bmatrix} \text{1 .. 99 Vector[column]} \\ \text{Data Type: float[8]} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

```
> Z3 := evalf(Newton_update(Z2));
```

$$Z3 := \begin{bmatrix} \text{1 .. 99 Vector[column]} \\ \text{Data Type: float[8]} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

```
> Z4 := evalf(Newton_update(Z3));
```

$$Z4 := \begin{bmatrix} \text{1 .. 99 Vector[column]} \\ \text{Data Type: float[8]} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

```
> Z5 := evalf(Newton_update(Z4));
```

$$Z5 := \begin{bmatrix} \text{1 .. 99 Vector[column]} \\ \text{Data Type: float[8]} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

```
> Z6 := evalf(Newton_update(Z5));
```

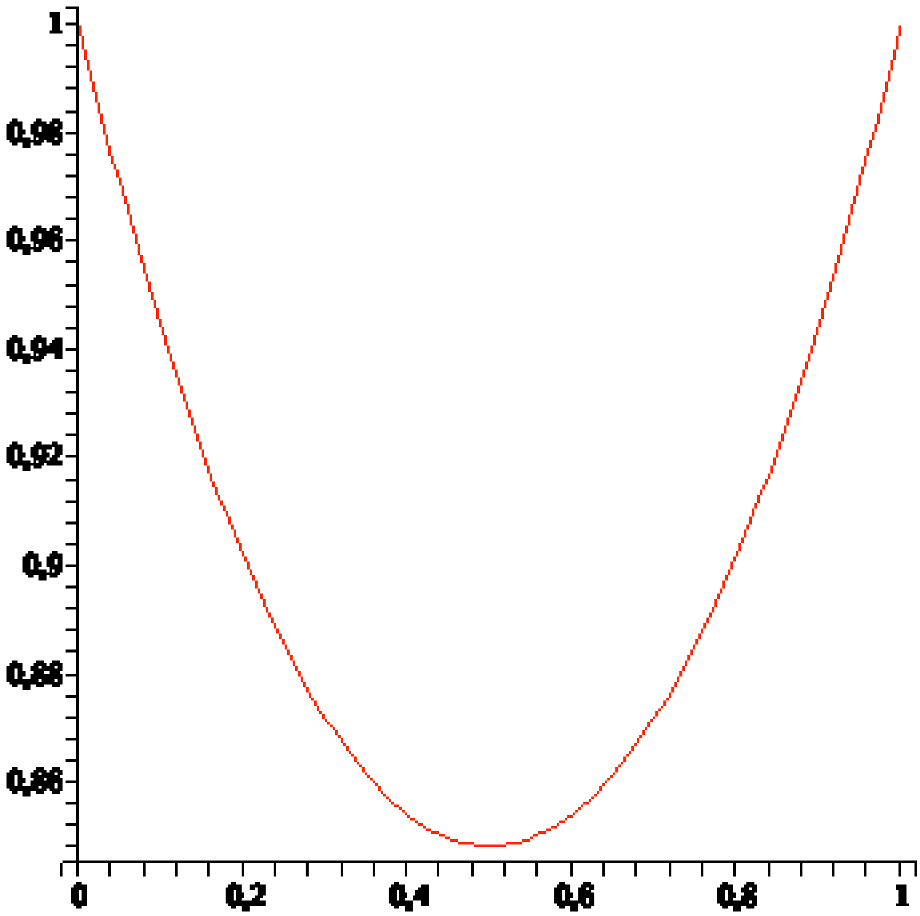
$$Z6 := \begin{bmatrix} \text{1 .. 99 Vector[column]} \\ \text{Data Type: float[8]} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

Verica del residuo di GRAD F

```
> resid := subs(seq(y[i]=Z6[i],i=1..n-1),y[0]=ya,y[n]=yb,eqns):
> Norm(Vector(resid),1);
```

0.00581452030000000151

```
> yyy := [1,seq(Z6[k],k=1..n-1),1]:
  xxx := [seq(k/n,k=0..n)]:
> plot([seq([xxx[k],yyy[k]],k=1..nops(yyy))]);
```




```
> lambda
```