Soluzioni del compito di

Metodi Matematici e Calcolo per Ingegneria

del 20 febbraio 2006

Enrico Bertolazzi

Trasformata di Laplace

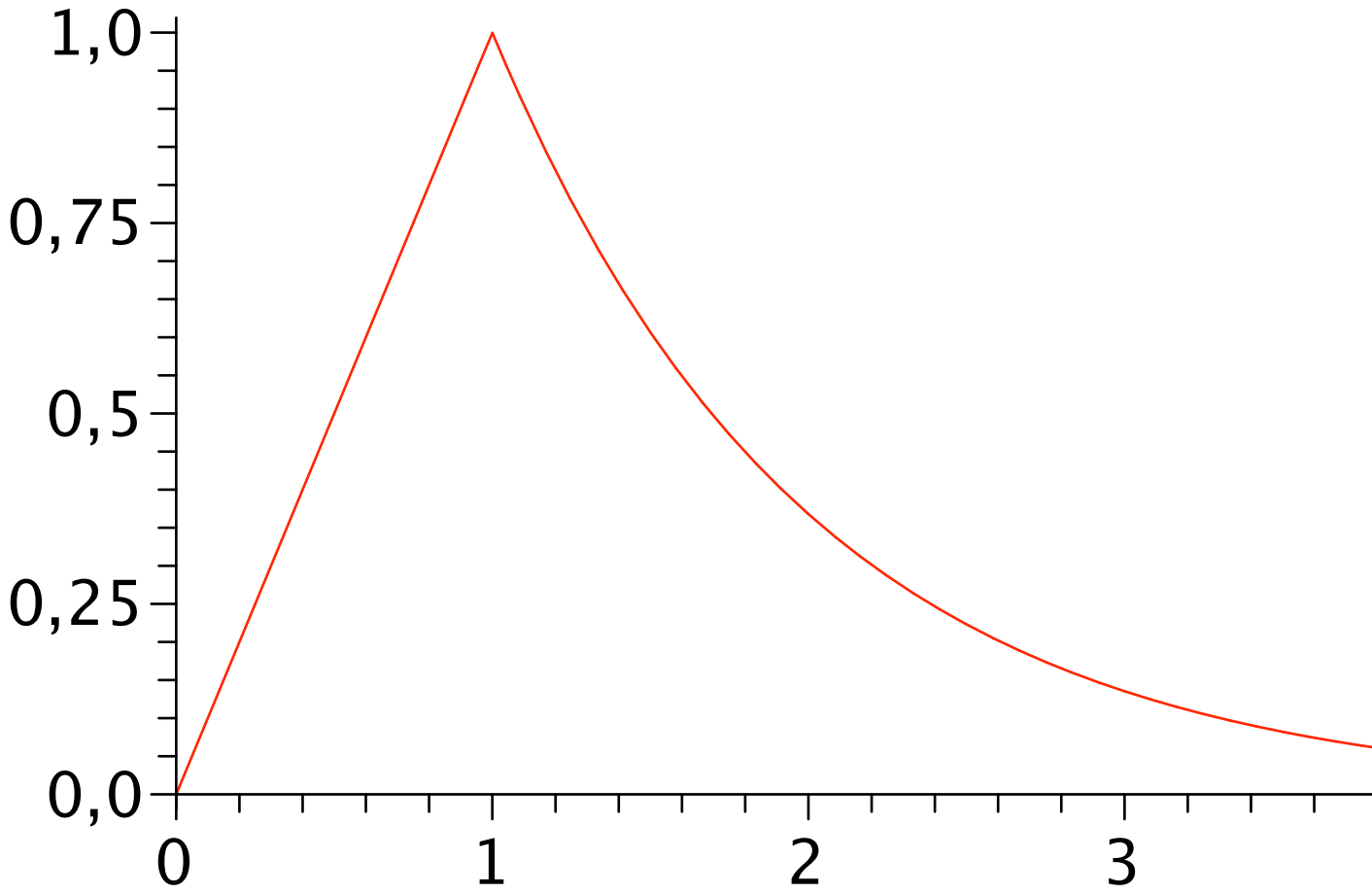
```
> restart:
  with(inttrans) :
```

Data la seguente funzione

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

$$f := t \mapsto t\,\theta(1-t)+e^{1-t}\,\theta(t-1)$$

```
> 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) ;
```

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

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

$$f3 := t \mapsto \theta(1-t) - t\,\delta(t-1) - e^{1-t}\,\theta(t-1) + e^{1-t}\,\delta(t-1)$$

Trasformate con le primitive Maple

```
> laplace(f(t),t,s);
  laplace(f1(t),t,s);
  laplace(f2(t),t,s);
  laplace(f3(t),t,s);
```

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

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

$$\frac{1-e^{-2\,s-4}}{2\,(s+2)^2}-\frac{e^{-2\,s-4}}{(s+2)\,(2\,s+5)}$$

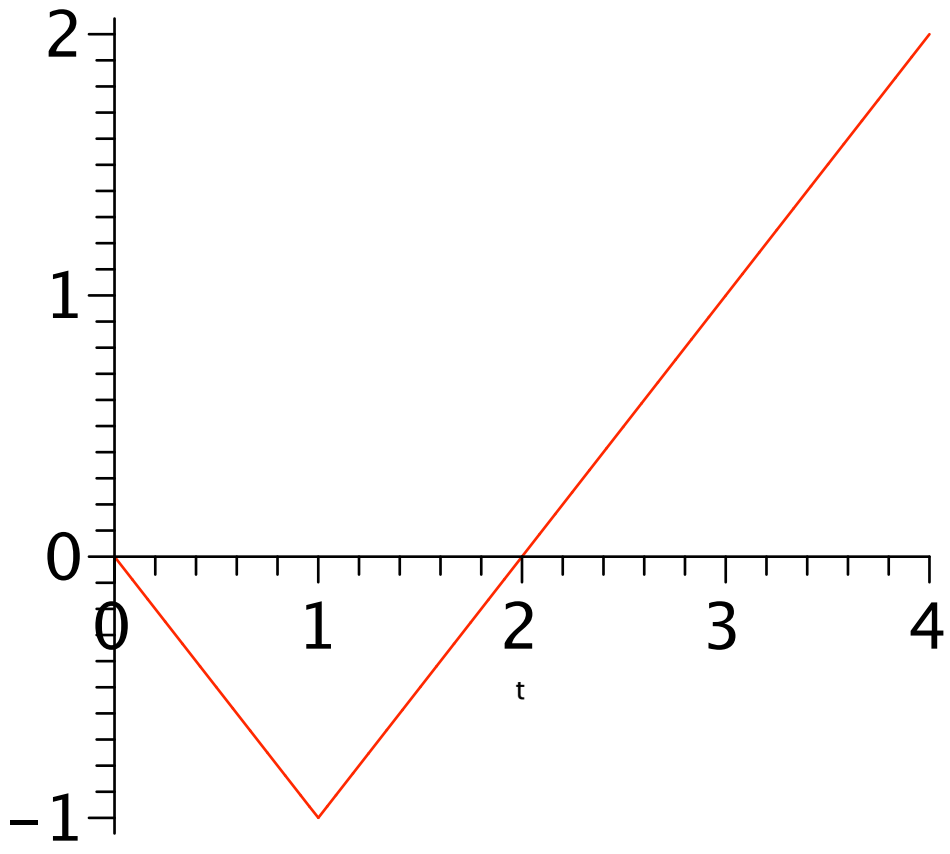
$$\frac{1+s-e^{-s}\,(2\,s+1)}{(s+1)\,s}$$

Soluzione di ODE con Laplace

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

$$src := t \mapsto -t+2\,(t-1)\,\theta(t-1)$$

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



Data la seguente equazione differenziale

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

$$ode := y'(x) = -x+2\,(x-1)\,\theta(x-1)$$

Con dato iniziale

```
> y0 := 10 ;
```

$$y0 := 10$$

Calcolare le soluzioni con le trasformate di Laplace.
Trasformo la equazione differenziale con la trasformata di Laplace

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

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

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

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

Risolvo la equazione per y(s)

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

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

Applico le condizioni iniziali ottenendo y(s)

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

$$ly := -\frac{1}{s^3} + \frac{2}{s^3\,e^s} + \frac{10}{s}$$

Espansione in fratti semplici

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

$$-\frac{1}{s^3} + \frac{2}{s^3\,e^s} + \frac{10}{s}$$

Antitrasformo per ottenere la equazione y(x)

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

$$res := -\frac{t^2}{2} + \theta(t-1)\,(t-1)^2 + 10$$

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);
```

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

```
> ode1 := 4*yp   -zp   -wp = 0 ;
  ode2 :=  -yp +4*zp   -wp = exp(-t) ;
  ode3 :=  -yp   -zp +4*wp = 0 ;
```

$$ode1 := 4\,\dot{y}(t) - \dot{z}(t) - \dot{w}(t) = 0$$

$$ode2 := -\dot{y}(t) + 4\,\dot{z}(t) - \dot{w}(t) = e^{-t}$$

$$ode3 := -\dot{y}(t) - \dot{z}(t) + 4\,\dot{w}(t) = 0$$

Con dato iniziale

```
> y0, z0, w0 := 3, 2, 1 ;
```

$$y0, z0, w0 := 3, 2, 1$$

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) ;
```

$$sode1 := 4\,s\,laplace(y(t),t,s) - 4\,y(0) - s\,laplace(z(t),t,s) + z(0) - s\,laplace(w(t),t,s) + w(0) = 0$$

$$sode2 := -s\,laplace(y(t),t,s) + y(0) + 4\,s\,laplace(z(t),t,s) - 4\,z(0) - s\,laplace(w(t),t,s) + w(0) = \frac{1}{1+s}$$

$$sode3 := -s\,laplace(y(t),t,s) + y(0) - s\,laplace(z(t),t,s) + z(0) + 4\,s\,laplace(w(t),t,s) - 4\,w(0) = 0$$

```
> 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);
```

$$4\,s\,y(s) - 4\,y(0) - s\,z(s) + z(0) - s\,w(s) + w(0) = 0$$

$$-s\,y(s) + y(0) + 4\,s\,z(s) - 4\,z(0) - s\,w(s) + w(0) = \frac{1}{1+s}$$

$$-s\,y(s) + y(0) - s\,z(s) + z(0) + 4\,s\,w(s) - 4\,w(0) = 0$$

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});
```

$$RES := \left\{ laplace(w(t),t,s) = \frac{10\,w(0) + 10\,w(0)\,s + 1}{10\,s\,(1+s)},\ laplace(z(t),t,s) = \frac{3 + 10\,z(0) + 10\,z(0)\,s}{10\,s\,(1+s)},\ laplace(y(t),t,s) = \frac{1 + 10\,y(0) + 10\,y(0)\,s}{10\,s\,(1+s)} \right\}$$

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

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

$$SOL := \begin{bmatrix} \frac{31+30\,s}{10\,s\,(1+s)} \\ \frac{23+20\,s}{10\,s\,(1+s)} \\ \frac{11+10\,s}{10\,s\,(1+s)} \end{bmatrix}$$

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) ;
```

$$yy := -\frac{e^{-x}}{10} + \frac{31}{10}$$

$$zz := -\frac{3\,e^{-x}}{10} + \frac{23}{10}$$

$$ww := -\frac{e^{-x}}{10} + \frac{11}{10}$$

Espansione in fratti semplici per controllo

```
> convert(SOL[1], fullparfrac, s);
  convert(SOL[2], fullparfrac, s);
  convert(SOL[3], fullparfrac, s);
```

$$\frac{31}{10\,s} - \frac{1}{10\,(1+s)}$$

$$\frac{23}{10\,s} - \frac{3}{10\,(1+s)}$$

$$\frac{11}{10\,s} - \frac{1}{10\,(1+s)}$$

Soluzione di ricorrenza con trasformata zeta

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

Risolvere la seguente ricorrenza

```
> RIC := f(n+2) = 2*f(n+1) + 3*f(n) - n ;
```

$$RIC := f(n+2) = 2\,f(n+1) + 3\,f(n) - n$$

Con dato iniziale

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

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

Usando le primitive di maple:

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

$$-\frac{3\,(-1)^k}{16}+\frac{3\,3^k}{16}+\frac{k}{4}$$

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

$$\frac{3\,(-1)^{k+1}}{16}+\frac{3^{k+1}}{16}+\frac{k}{4}$$

Usando la Z-trasformata

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

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

Ricavo f(z)

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

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

Applico le condizioni iniziali

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

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

Conversione in fratti semplici di f(z)/z

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

$$\frac{\mathrm{ztrans}(f(n),n,z)}{z}=\frac{3}{16\,(z-3)}+\frac{1}{4\,(z-1)^2}-\frac{3}{16\,(z+1)}$$

```
> invztrans(%*z,z,k) ;
```

$$f(k)=-\frac{3\,(-1)^k}{16}+\frac{3\,3^k}{16}+\frac{k}{4} \qquad (4.1)$$

Inversione della Z-trasformata

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

$$f(k)=-\frac{3\,(-1)^k}{16}+\frac{3\,3^k}{16}+\frac{k}{4}$$

Soluzione di un sistema non lineare con Newton

```
> restart:
  with(VectorCalculus):
```

Sistema non lineare

```
> f := 4*x-y + x*y + 1 ;
  g :=     x + 2*y - x*y - 2 ;
```

$$f := 4\,x-y+x\,y+1$$

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

Soluzione esatta

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

$$\{y=1,\ x=0\},\ \{x=2,\ y=-9\}$$

Matrice Jacobiano

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

$$J := \begin{bmatrix} 4+y & -1+x \\ 1-y & -x+2 \end{bmatrix}$$

Schema di Newton

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

$$Newton\_update := -\frac{x\,(-1+y)}{5\,x-9-y}\,e_x + \left(\frac{5\,x\,y-y-9}{5\,x-9-y}\right)e_y$$

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])) ;
```

$$x_{k+1} = -\frac{x_k\left(-1+y_k\right)}{5\,x_k-9-y_k}$$

$$y_{k+1} = \frac{5\,x_k y_k-y_k-9}{5\,x_k-9-y_k}$$

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])) ;
```

$$x_1 := 0.1666666667$$

$$y_1 := 0.1666666667$$

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])) ;
```

$$x_2 := -0.01666666667$$

$$y_2 := 1.083333333$$

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])) ;
```

$$x_3 := -0.0001366120213$$

$$y_3 := 1.000683060$$

Problema di Minimo Vincolato

```
> restart:
  with(LinearAlgebra):
  with(Optimization):
  with(VectorCalculus):
```

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)*(y+z)=1,x-z=1] ;
```

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

N OTA: lo spazio dei voncoli sono due piani intsersecanti!

```
> solve(expand(subs(z=x-1,(x+y)*(y+z)=1)),x) ;
```

$$\frac{1}{2}+\frac{\sqrt{5}}{2}-y,\ \frac{1}{2}-\frac{\sqrt{5}}{2}-y$$

Soluzione con le primitive Maple

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

[1.49999999999999956,

$$[x = 1.05901699412087579,\ y = 0.559016994629018882,\ z = 0.0590169941208759444]$$

]

Uso dei moltiplicatori di Lagrange

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

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

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

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

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

Sistema non lineare da risolvere

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

$$F := (4x-2y-2z-\lambda\ (y+z)-\mu)\bar{e}_x+(-2x+4y-2z-\lambda\ (2y+z+x))\bar{e}_y$$

$$+(-2x+4z-2y-\lambda\ (x+y)+\mu)\bar{e}_z+(-(x+y)\ (y+z)+1)\bar{e}_\lambda+(-x+z+1)\bar{e}_\mu$$

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})]));
```

$$RES := \left\{y = \frac{\sqrt{5}}{4},\ \lambda = 0,\ \mu = 3,\ z = -\frac{1}{2}+\frac{\sqrt{5}}{4},\ x = \frac{1}{2}+\frac{\sqrt{5}}{4}\right\},\ \left\{y = -\frac{\sqrt{5}}{4},\ \lambda = 0,\ \mu = 3,\right.$$

$$z = -\frac{1}{2} - \frac{\sqrt{5}}{4},\ x = \frac{1}{2} - \frac{\sqrt{5}}{4}\Big\}$$

Prima soluzione

```
> RES[1];
```

$$\left\{y = \frac{\sqrt{5}}{4},\ \lambda = 0,\ \mu = 3,\ z = -\frac{1}{2} + \frac{\sqrt{5}}{4},\ x = \frac{1}{2} + \frac{\sqrt{5}}{4}\right\}$$

Seconda soluzione

```
> RES[2];
```

$$\left\{y = -\frac{\sqrt{5}}{4},\ \lambda = 0,\ \mu = 3,\ z = -\frac{1}{2} - \frac{\sqrt{5}}{4},\ x = \frac{1}{2} - \frac{\sqrt{5}}{4}\right\}$$

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 ;
```

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

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

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

$$NH := \left\{\begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}\right\}$$

Controllo minimo/massimo locale primo punto

```
> lambda1 := subs(RES[1],lambda);
  mu1     := subs(RES[1],mu);
```

$$\lambda 1 := 0$$

$$\mu 1 := 3$$

Calcolo l'Hessiano nel punto stazionario

```
> Hf1 := simplify(subs(RES[1],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(RES[1],op(NH)) ;
```

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

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

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

$$16\,\alpha^2$$

```
> subs(RES[1],f);
```

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

Controllo minimo/massimo locale secondo punto

```
> lambda2 := subs(RES[2],lambda);
  mu2     := subs(RES[2],mu);
```

$$\lambda 2 := 0$$

$$\mu 2 := 3$$

```
> Hf2 := simplify(subs(RES[2],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(RES[2],op(NH)) ;
```

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

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

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

$$16\,\alpha^2$$

```
> subs(RES[2],f);
```

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

Approssimazione di un problema del calcolo delle variazioni

```
> restart:
```

Integrale da minimizzare

```
> int(y(x)*(1+diff(y(x),x)^2),x=0..1);
```

$$\int_0^1 y(x)\left(1+y'(x)^2\right)\,dx$$

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);
```

$$F := \frac{(y_1+y_0)\left(1+16\,(y_1-y_0)^2\right)}{2}+\frac{(y_2+y_1)\left(1+16\,(y_2-y_1)^2\right)}{2}+$$

$$\frac{(y_3+y_2)\left(1+16\,(y_3-y_2)^2\right)}{2}+\frac{(y_4+y_3)\left(1+16\,(y_4-y_3)^2\right)}{2}$$

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

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

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

$$eqns\_fun := (y\_1, y\_2, y\_3) \mapsto \mathrm{rtable}\Big(1..3,$$

$$\{1 = -7+48\,y\_1^2-16\,y\_1-8\,y\_2^2-16\,y\_2\,y\_1,$$

$$2 = 1+48\,y\_2^2-16\,y\_2\,y\_1-8\,y\_1^2-8\,y\_3^2-16\,y\_3\,y\_2,$$

$$3 = -7+48\,y\_3^2-16\,y\_3\,y\_2-8\,y\_2^2-16\,y\_3\},\ datatype = anything,$$

$$subtype = Vector_{column},\ storage = rectangular,\ order = Fortran\_order,$$

$$attributes = [coords = cartesian]\Big)$$

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

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

$$J := (x, y, z) \mapsto \mathrm{rtable}(1..3, 1..3,$$
$$\{(1,1) = 96x - 16 - 16y,\ (1,2) = -16y - 16x,\ (2,1) = -16y - 16x,$$
$$(2,2) = 96y - 16x - 16z,\ (2,3) = -16z - 16y,\ (3,2) = -16z - 16y,$$
$$(3,3) = 96z - 16y - 16\},\ datatype = anything,\ subtype = Matrix,$$
$$storage = rectangular,\ order = Fortran\_order)$$

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

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

```
> 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));
```

$$Z2 := (0.9512040199000000030)e_x + (0.934561965500000035)e_y$$
$$+ (0.9512040199000000030)e_z$$

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

$$Z3 := (0.9512004214000000051)e_x + (0.934555355300000001)e_y$$
$$+ (0.9512004214000000051)e_z$$

Verica del residuo di GRAD F

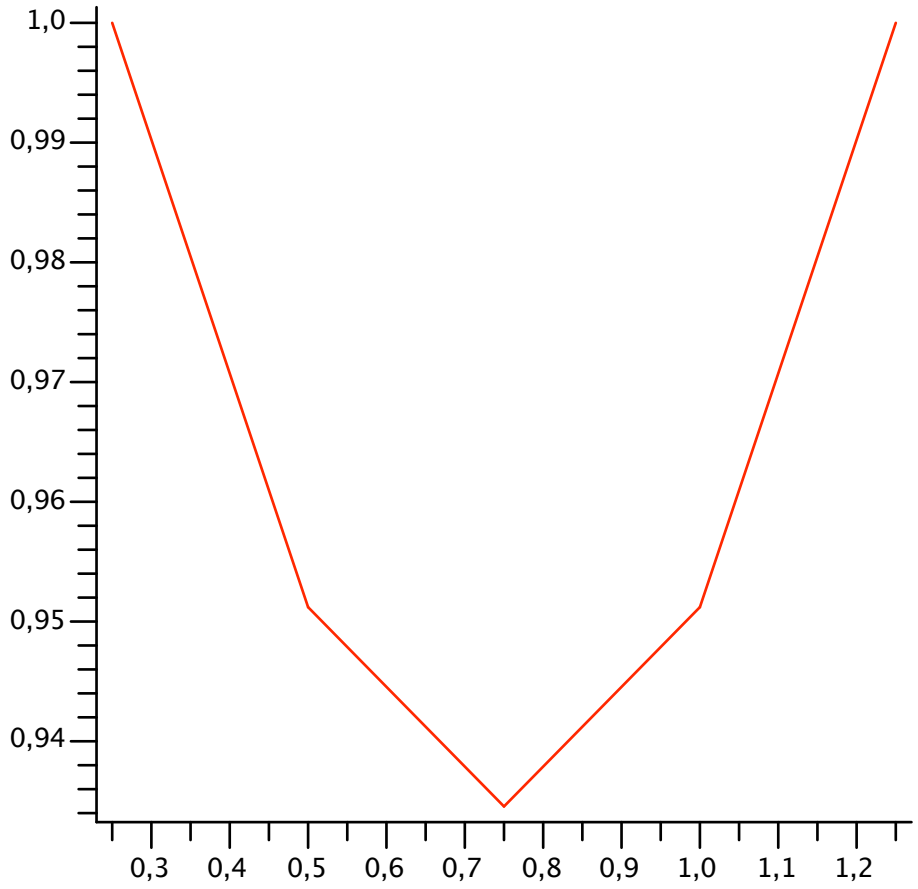
```
> subs(seq(y[i]=Z3[i],i=1..n-1),y[0]=ya,y[n]=yb,eqns) ;
```

$$[-1.3\ 10^{-8}, -4.2\ 10^{-9}, -1.3\ 10^{-8}, 0, 0]$$

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

$$yyy := [1, 0.9512004214000000051, 0.934555355300000001, 0.9512004214000000051, 1]$$

```
> plot([seq([k*h,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,symbolic),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)));
```

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

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

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

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

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

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

$$Z2 := \begin{bmatrix} 1..99\ Vector_{column} \\ Data\ Type: float_8 \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix}$$

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

$$Z3 := \begin{bmatrix} 1..99\ Vector_{column} \\ Data\ Type: float_8 \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix}$$

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

$$Z4 := \begin{bmatrix} 1 .. 99\ Vector_{column} \\ Data\ Type: float_8 \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix}$$

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

$$Z5 := \begin{bmatrix} 1 .. 99\ Vector_{column} \\ Data\ Type: float_8 \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix}$$

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

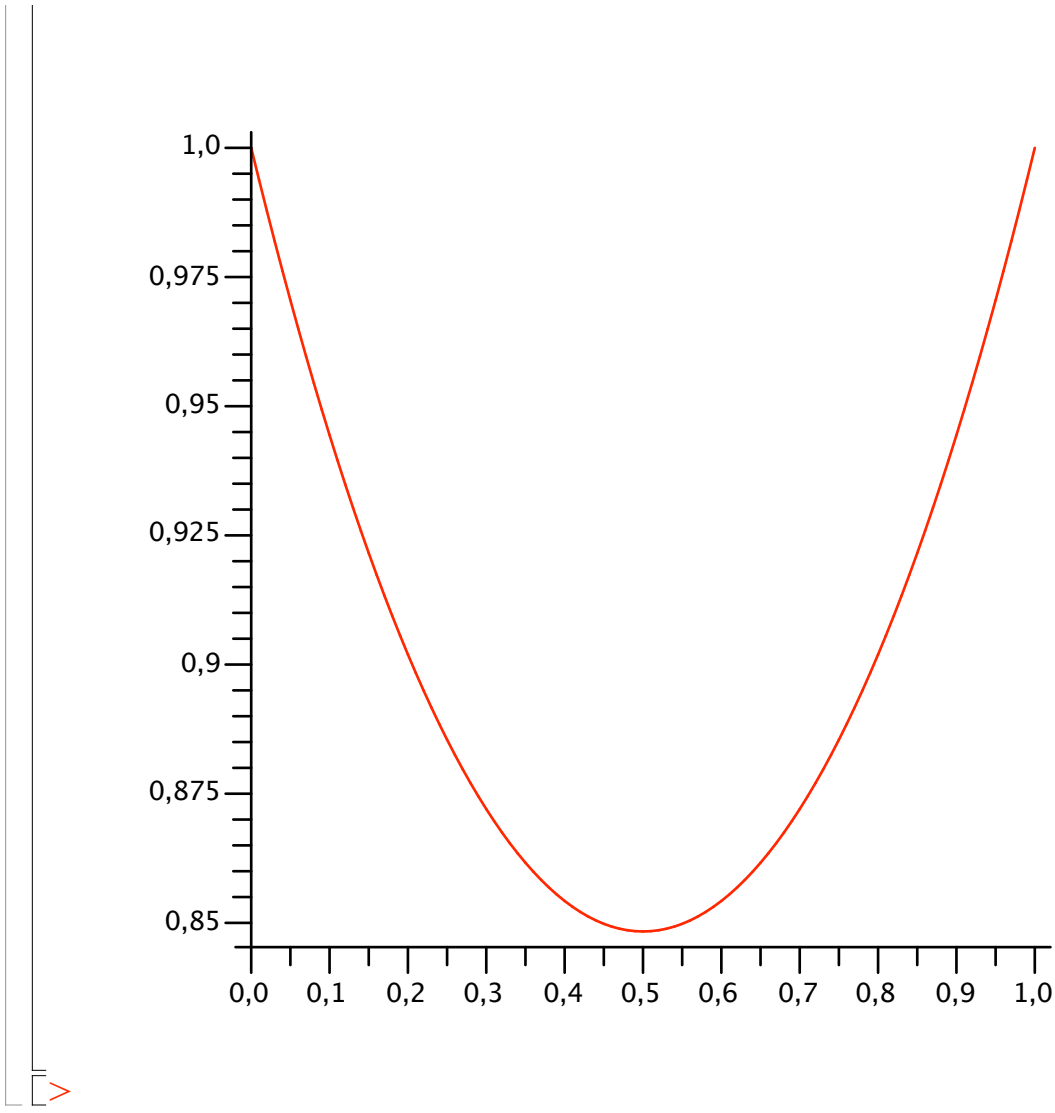
$$Z6 := \begin{bmatrix} 1 .. 99\ Vector_{column} \\ Data\ Type: float_8 \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix}$$

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);
```

0.00621011449999999635

```
> 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))]);
```


1,0
0,975
0,95
0,925
0,9
0,875
0,85
0,0
0,1
0,2
0,3
0,4
0,5
0,6
0,7
0,8
0,9
1,0