Soluzioni del compito di

Metodi Matematici e Calcolo per Ingegneria

del 30 agosto 2006

Enrico Bertolazzi

Trasformata di Laplace

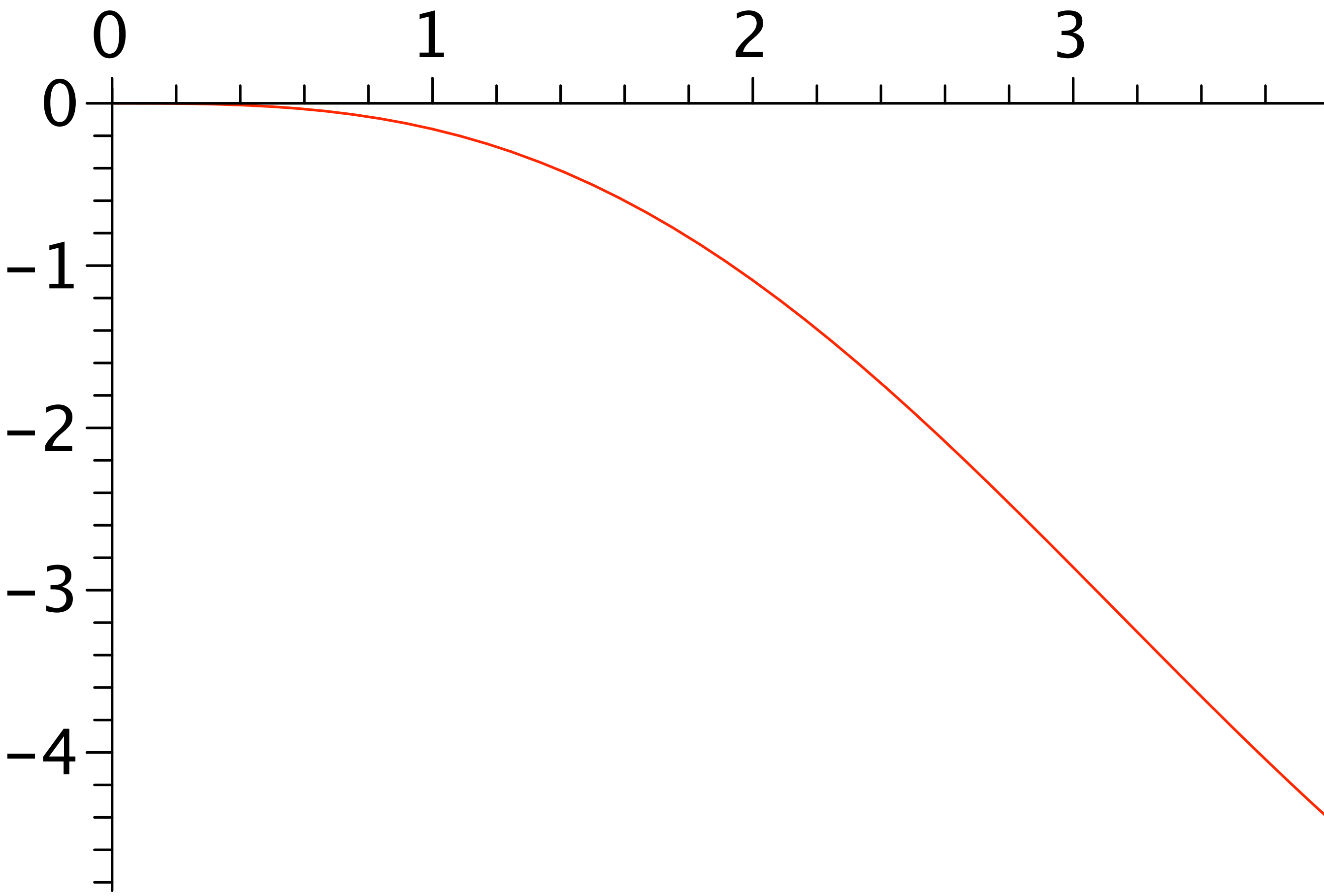
```
> restart:
  with(inttrans) :
```

Data la seguente funzione

```
> f := t -> sin(t)-t ;
```

$$f := t \mapsto \sin(t) - t$$

```
> `diff/f` := proc() 1 ; end:
> 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) ;
```

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

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

$$f3 := t \mapsto \cos(t) - 1$$

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);
```

$$-\frac{1}{\left(s^2+1\right) s^2}$$

$$-\frac{1}{3\left(9 s^2+1\right) s^2}$$

$$-\frac{1}{2\left(4 s^2+16 s+17\right)(s+2)^2}$$

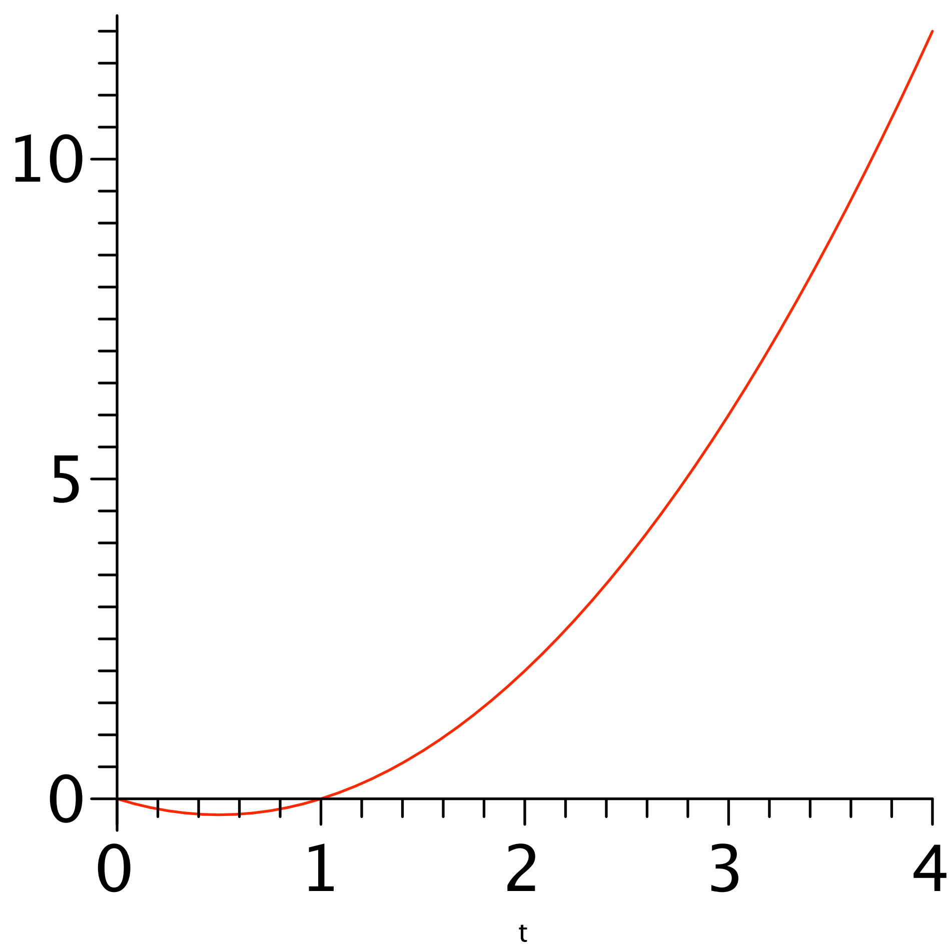
$$-\frac{1}{\left(s^2+1\right) s}$$

Soluzione di ODE con Laplace

```
> restart:
  with(inttrans) :
> src := t -> t^2-t ;
```

$$src := t \mapsto t^2 - t$$

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



Data la seguente equazione differenziale

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

$$ode := y''(x) - y'(x) = x^2 - x$$

Con dato iniziale

```
> y0,yp0 := 10,-10 ;
```

$$y0, yp0 := 10, -10$$

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

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

$$sode := s^2\, laplace(y(x), x, s) - D(y)(0) - s\, y(0) - s\, laplace(y(x), x, s) + y(0) = \frac{2}{s^3} - \frac{1}{s^2}$$

```
> sode1 := subs(laplace(y(x),x,s)=y(s),y(0)=y0,D(y)(0)=yp0,sode) ;
```

$$sode1 := s^2\, y(s) + 20 - 10\, s - s\, y(s) = \frac{2}{s^3} - \frac{1}{s^2}$$

Risolvo la equazione per y(s)

```
> lode := isolate(sode1,y(s));
```

$$lode := y(s) = \frac{\frac{2}{s^3} - \frac{1}{s^2} - 20 + 10\, s}{s^2 - s}$$

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

$$y(s) = \frac{2 - s - 20\, s^3 + 10\, s^4}{s^4\, (s-1)}$$

Applico le condizioni iniziali ottenendo y(s)

```
> ly := expand(subs( y(0)=y0,D(y)(0)=yp0,rhs(lode))) ;
```

$$ly := \frac{2}{(s^2-s)\, s^3} - \frac{1}{(s^2-s)\, s^2} - \frac{20}{s^2-s} + \frac{10\, s}{s^2-s}$$

Espansione in fratti semplici

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

$$-\frac{9}{s-1} - \frac{1}{s^2} - \frac{2}{s^4} - \frac{1}{s^3} + \frac{19}{s}$$

Antitrasformo per ottenere la equazione y(x)

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

$$res := -\frac{t^2}{2} - t - \frac{t^3}{3} - 9\, e^t + 19$$

Soluzione di un sistema di ODE con Laplace

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

Dato il seguente sistema di equazioni differenziali

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

$$yp, zp, wp := y'(x), z'(x), w'(x)$$

```
> ode1 := 3*yp -2*zp   -wp = x ;
  ode2 :=  -yp   +zp   -wp = 2*x ;
  ode3 :=  -yp -2*zp +3*wp = 3*x ;
```

$$ode1 := 3\, y'(x) - 2\, z'(x) - w'(x) = x$$

$$ode2 := -y'(x) + z'(x) - w'(x) = 2x$$

$$ode3 := -y'(x) - 2z'(x) + 3w'(x) = 3x$$

Con dato iniziale

```
> y0, z0, w0 := 1, 0, 1 ;
```

$$y0, z0, w0 := 1, 0, 1$$

```
> INI := y(0)=y0,z(0)=z0,w(0)=w0 ;
```

$$INI := y(0) = 1, z(0) = 0, w(0) = 1$$

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

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

$$sode1 := 3\,s\,laplace(y(x), x, s) - 2 - 2\,s\,laplace(z(x), x, s) - s\,laplace(w(x), x, s) = \frac{1}{s^2}$$

$$sode2 := -s\,laplace(y(x), x, s) + 2 + s\,laplace(z(x), x, s) - s\,laplace(w(x), x, s) = \frac{2}{s^2}$$

$$sode3 := -s\,laplace(y(x), x, s) - 2 - 2\,s\,laplace(z(x), x, s) + 3\,s\,laplace(w(x), x, s) = \frac{3}{s^2}$$

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

$$3\,s\,y(s) - 2 - 2\,s\,z(s) - s\,w(s) = \frac{1}{s^2}$$

$$-s\,y(s) + 2 + s\,z(s) - s\,w(s) = \frac{2}{s^2}$$

$$-s\,y(s) - 2 - 2\,s\,z(s) + 3\,s\,w(s) = \frac{3}{s^2}$$

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

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

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

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

$$RES := \left\{ laplace(z(x), x, s) = -\frac{4}{s^3}, laplace(y(x), x, s) = \frac{4\,s^2 - 13}{4\,s^3}, \right.$$

$$laplace(w(x), x, s) = \frac{4\,s^2 - 11}{4\,s^3}\Big\}$$

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

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

$$SOL := \begin{bmatrix} \frac{4\,s^2 - 13}{4\,s^3} \\ -\frac{4}{s^3} \\ \frac{4\,s^2 - 11}{4\,s^3} \end{bmatrix}$$

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

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

$$yy := 1 - \frac{13\,x^2}{8}$$

$$zz := -2\,x^2$$

$$ww := 1 - \frac{11\,x^2}{8}$$

Espansione in fratti semplici per controllo

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

$$-\frac{13}{4\,s^3} + \frac{1}{s}$$

$$-\frac{4}{s^3}$$

$$-\frac{11}{4\,s^3} + \frac{1}{s}$$

Soluzione di ricorrenza con trasformata zeta

```
> restart:
```

Risolvere la seguente ricorrenza

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

$$RIC := f(n+2) = f(n) + n + 1$$

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));
```

$$\frac{3}{8}-\frac{7\,(-1)^k}{8}+\frac{(k+1)\left(\frac{k}{2}+1\right)}{2}-\frac{3\,k}{4}$$

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

$$\frac{7}{8}+\frac{7\,(-1)^{k+1}}{8}+\frac{k^2}{4}$$

Usando la Z-trasformata

```
> zRIC := subs(INI,ztrans(f(n),n,z)=f(z),ztrans(RIC,n,z));#
```

$$zRIC := z^2 f(z)-2\,z=f(z)+\frac{z}{(z-1)^2}+\frac{z}{z-1}$$

Ricavo f(z)

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

$$zRICrhs := f(z)=\frac{2\,z+\frac{z}{(z-1)^2}+\frac{z}{z-1}}{z^2-1}$$

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

$$f(z)=\frac{z\left(2\,z^2-3\,z+2\right)}{(z-1)^2\left(z^2-1\right)}$$

Conversione in fratti semplici di f(z)/z

```
> expr := convert(rhs(zRICrhs)/z, parfrac);
```

$$expr := \frac{1}{4\,(z-1)^2}+\frac{1}{2\,(z-1)^3}+\frac{7}{8\,(z-1)}-\frac{7}{8\,(z+1)}$$

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

$$\frac{7}{8}-\frac{7\,(-1)^k}{8}+\frac{k^2}{4} \qquad (4.1)$$

Inversione della Z-trasformata (controllo con MAPLE)

```
> invztrans(rhs(zRICrhs),z,k) ;
```

$$\frac{7}{8}-\frac{7\,(-1)^k}{8}+\frac{k^2}{4}$$

Soluzione di un sistema non lineare con Newton

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

Sistema non lineare

```
> f := 3*x + y   + x/y + 1 ;
  g :=   x + 2*y - x/y - 1 ;
```

$$f := 3\,x+y+\frac{x}{y}+1$$

$$g := x + 2\,y - \frac{x}{y} - 1$$

Soluzione esatta

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

$$\left\{y = \frac{1}{5},\ x = -\frac{3}{20}\right\}$$

Matrice Jacobiano

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

$$J := \begin{bmatrix} 3 + \frac{1}{y} & 1 - \frac{x}{y^2} \\ 1 - \frac{1}{y} & 2 + \frac{x}{y^2} \end{bmatrix}$$

Schema di Newton

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

$$Newton\_update := -\frac{3\,y\,(x+y)}{5\,y^2 + 4\,x + 3\,y}e_x + \frac{4\,y\,(x+y)}{5\,y^2 + 4\,x + 3\,y}e_y$$

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])) ;
```

$$x_{k+1} = -\frac{3\,y_k\,(x_k + y_k)}{5\,y_k^2 + 4\,x_k + 3\,y_k}$$

$$y_{k+1} = \frac{4\,y_k\,(x_k + y_k)}{5\,y_k^2 + 4\,x_k + 3\,y_k}$$

Tre iterate a partire da (1,2)

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

$$x_0, y_0 := 0, 2$$

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])) ;
```

$$x_1 := -0.4615384615$$

$$y_1 := 0.6153846154$$

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])) ;
```

$$x_2 := -0.1500000001$$

$$y_2 := 0.2000000001$$

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])) ;
```
$$x_3 := -0.1499999999$$

$$y_3 := 0.1999999999$$

Problema di Minimo Vincolato

```
> restart:
  with(LinearAlgebra):
  with(Optimization):
  with(VectorCalculus):
```
Minimizzare la seguente funzione
```
> f := x*z/y;
```
$$f := \frac{x\,z}{y}$$

Soggetta ai vicoli
```
> v := [x+y+z=1,x^2+y^2+z^2=1] ;
```
$$v := [x + y + z = 1, x^2 + y^2 + z^2 = 1]$$

Soluzione con le primitive Maple
```
> Minimize(f, v );
```
$$[2.77333911991761964\ 10^{-32},$$

$$[x = -1.66533453693773481\ 10^{-16}, z = -1.66533453693773481\ 10^{-16}, y = 1.]]$$

Uso dei moltiplicatori di Lagrange
```
> v1 := lhs(v[1])-rhs(v[1]) ;
  v2 := lhs(v[2])-rhs(v[2]) ;
```
$$v1 := x + y + z - 1$$

$$v2 := x^2 + y^2 + z^2 - 1$$

```
> g := f - lambda*v1 - mu*v2 ;
```
$$g := \frac{x\,z}{y} - \lambda\,(x + y + z - 1) - \mu\,(x^2 + y^2 + z^2 - 1)$$

Sistema non lineare da risolvere
```
> F := Gradient(g,[x,y,z,lambda,mu]) ;
```
$$F := \left(\frac{z}{y} - \lambda - 2\,\mu\,x\right)\bar{e}_x + \left(-\frac{x\,z}{y^2} - \lambda - 2\,\mu\,y\right)\bar{e}_y + \left(\frac{x}{y} - \lambda - 2\,\mu\,z\right)\bar{e}_z + (-x - y - z + 1)\bar{e}_\lambda$$

$$+ \left(-x^2 - y^2 - z^2 + 1\right)\bar{e}_\mu$$

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})]));
```
$$RES := \left\{y = -\frac{1}{3}, \mu = 1, z = \frac{2}{3}, \lambda = -\frac{10}{3}, x = \frac{2}{3}\right\}, \{y = 1, \mu = 0, z = 0, \lambda = 0, x = 0\}$$

Prima soluzione

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

$$\left\{y=-\frac{1}{3}, \mu=1, z=\frac{2}{3}, \lambda=-\frac{10}{3}, x=\frac{2}{3}\right\}$$

Seconda soluzione

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

$$\{y=1, \mu=0, z=0, \lambda=0, x=0\}$$

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 ;
```

$$\begin{bmatrix} 0 & -\frac{z}{y^2} & \frac{1}{y} \\ -\frac{z}{y^2} & \frac{2\,x\,z}{y^3} & -\frac{x}{y^2} \\ \frac{1}{y} & -\frac{x}{y^2} & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

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

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

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

Controllo minimo/massimo locale primo punto

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

$$\lambda 1 := -\frac{10}{3}$$

$$\mu 1 := 1$$

Calcolo l'Hessiano nel punto stazionario

```
> Hf1 := simplify(subs(RES[1],Hf - lambda1. Hv1 - mu1. Hv2)) ;
```

$$Hf1 := \begin{bmatrix} -2 & -6 & -3 \\ -6 & -26 & -6 \\ -3 & -6 & -2 \end{bmatrix}$$

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

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

$$\begin{bmatrix} 1. \\ -2. \\ -29. \end{bmatrix}$$

Cerco nello spazio dei vincoli:

```
> Z1 := subs(RES[1],op(NH)) ;
```

$$Z1 := \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

E' negativo per ogni alpha, quindi è un massimo locale

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

$$2\,\alpha^2$$

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

$$-\frac{4}{3}$$

Controllo minimo/massimo locale secondo punto

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

$$\lambda 2 := 0$$

$$\mu 2 := 0$$

```
> Hf2 := simplify(subs(RES[2],Hf - lambda2. Hv1 - mu2. Hv2)) ;
```

$$Hf2 := \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

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

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

$$\begin{bmatrix} 0. \\ 1. \\ -1. \end{bmatrix}$$

Cerco nello spazio dei vincoli:

```
> Z2 := subs(RES[2],op(NH)) ;
```

$$Z2 := \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

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

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

$$-2\,\alpha^2$$

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

$$0$$

Approssimazione di un problema del calcolo delle variazioni

```
> restart:
```

Integrale da minimizzare

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

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

Condizioni al contorno

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

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

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

$$n := 4$$

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

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

```
> appF := ((y[k+1]+y[k])/2)+((y[k+1]-y[k])/h)^2 ;
```

$$appF := \frac{y_{k+1}}{2} + \frac{y_k}{2} + 16\left(y_{k+1} - y_k\right)^2$$

```
> F := sum( appF,k=0..n-1);
```

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

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

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

```
> res := fsolve({op(eqns)},{op(vars)}) ;
```

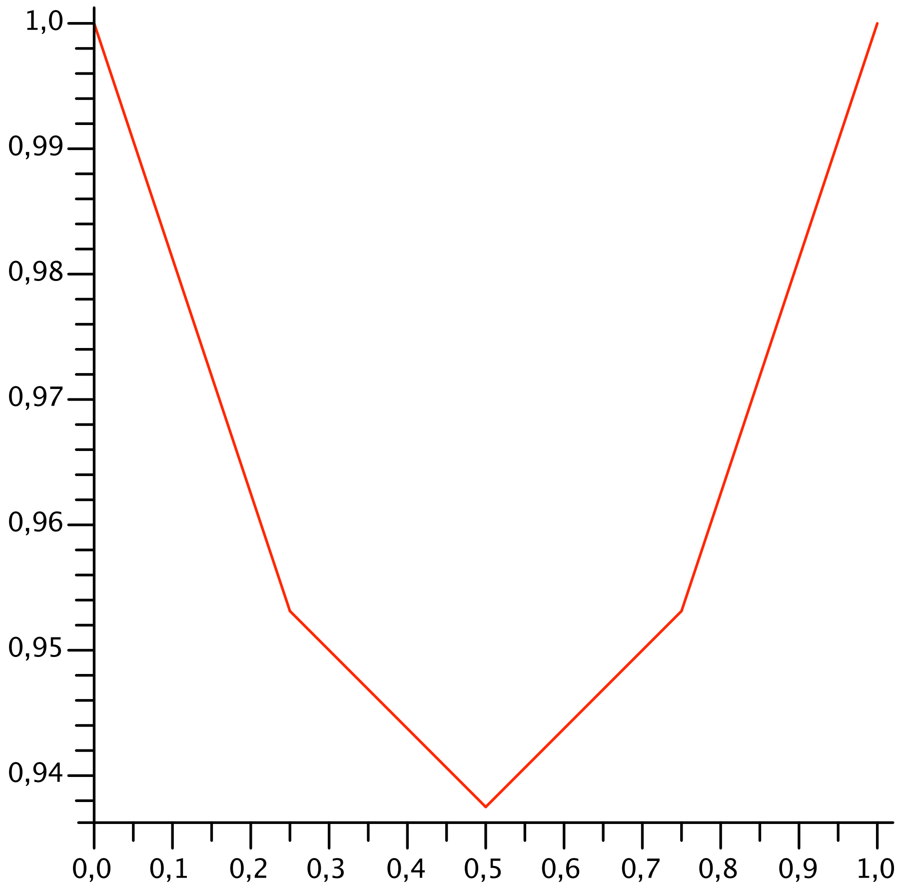
$$res := \left\{y_0 = 1., y_4 = 1., y_3 = 0.9531250000, y_1 = 0.9531250000, y_2 = 0.9375000000\right\}$$

```
> yy := subs(res,vars);
  xx := seq(k/n,k=0..n);
```

$$yy := [1., 0.9531250000, 0.9375000000, 0.9531250000, 1.]$$

$$xx := 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1$$

```
> plot([seq([xx[k],yy[k]],k=1..nops(yy))]);
```




```
> 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)];
```

$eqns\_fun := (y\_1, y\_2, y\_3)$

$\mapsto \mathrm{rtable}\Big(1..3,$

$\{1 = -31 + 64\, y\_1 - 32\, y\_2, 2 = 1 + 64\, y\_2 - 32\, y\_1 - 32\, y\_3,$

$3 = -31 + 64\, y\_3 - 32\, y\_2\}, datatype = anything, subtype = Vector_{column},$

$storage = rectangular, order = Fortran\_order, attributes = [coords = cartesian]\Big)$

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

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

$$J := (x, y, z) \mapsto \begin{bmatrix} 64 & -32 & 0 \\ -32 & 64 & -32 \\ 0 & -32 & 64 \end{bmatrix}$$

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

$Newton\_update := Z \rightarrow VectorCalculus{:-}{+}\left(Z, VectorCalculus{:-}{-}\left(LinearAlgebra{:-}LinearSolve\left(J\left(Z_1, Z_2, Z_3\right), eqns\_fun\left(Z_1, Z_2, Z_3\right)\right)\right)\right)$

```
> 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));
```

$Z2 := (0.953125000000000000)e_x + (0.937499999999999999)e_y + (0.953125000000000000)e_z$

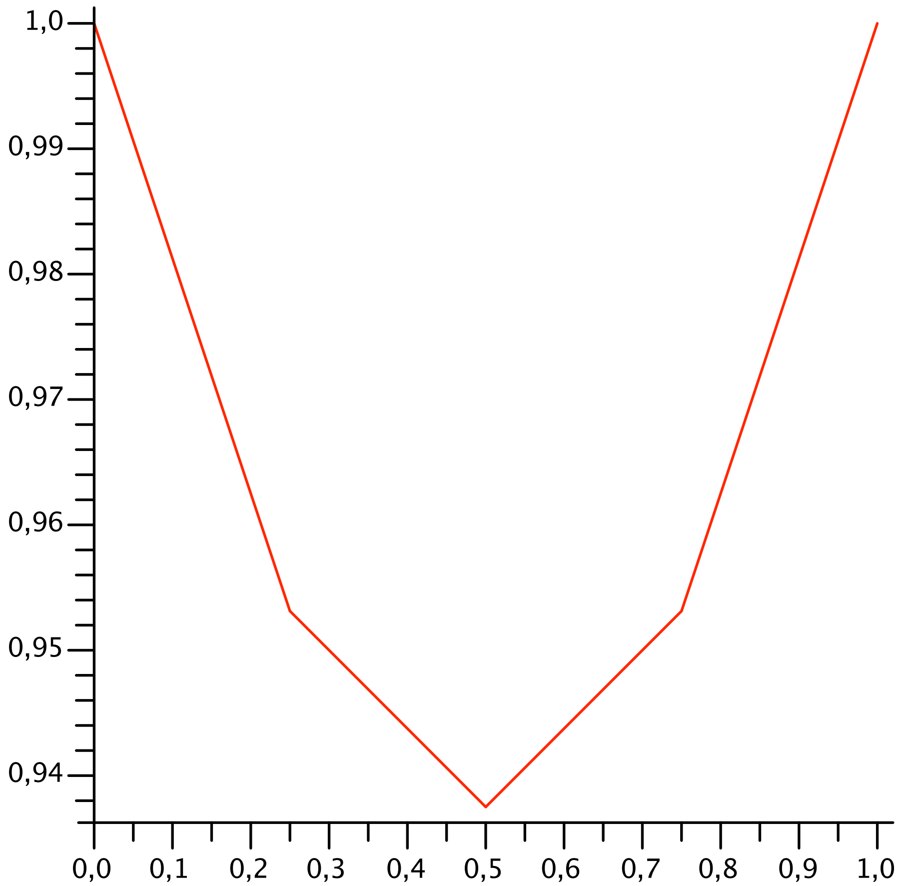
```
> Z3 := evalf(Newton_update(Z2));
```

$Z3 := (0.953125000000000000)e_x + (0.937499999999999999)e_y + (0.953125000000000000)e_z$

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

$yyy := [1, 0.953125000000000000, 0.937499999999999999, 0.953125000000000000, 1]$

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




```
> # Newton con molti piu punti!
> n := 50 ;
```

$$n := 50$$

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

$$h := \frac{1}{50}$$

```
> F := sum( appF,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

$$:= [y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}, y_{12}, y_{13}, y_{14}, y_{15}, y_{16}, y_{17}, y_{18}, y_{19}, y_{20}, y_{21}, y_{22}, y_{23}, y_{24}, y_{25}, y_{26}, y_{27}, y_{28}, y_{29}, y_{30}, y_{31}, y_{32}, y_{33}, y_{34}, y_{35}, y_{36}, y_{37}, y_{38}, y_{39}, y_{40}, y_{41}, y_{42}, y_{43}, y_{44}, y_{45}, y_{46}, y_{47}, y_{48}, y_{49}, y_{50}]$$

```
> 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* → *VectorCalculus*:-

$+\Big(Z$

, *VectorCalculus*:--(*LinearAlgebra*:-

$LinearSolve\Big(J\big(seq\big(Z_k, k = 1 ..VectorCalculus:- + (n, VectorCalculus:--(1))\big)\big),$

$eqns\_fun\big(seq\big(Z_k, k = 1 ..VectorCalculus:- + (n, VectorCalculus:--(1))\big)\big)\Big)\Big)\Big)$

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

$$Z0 := \begin{bmatrix} \text{1 .. 49 } Vector_{column} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

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

$$Z1 := \begin{bmatrix} \text{1 .. 49 } Vector_{column} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

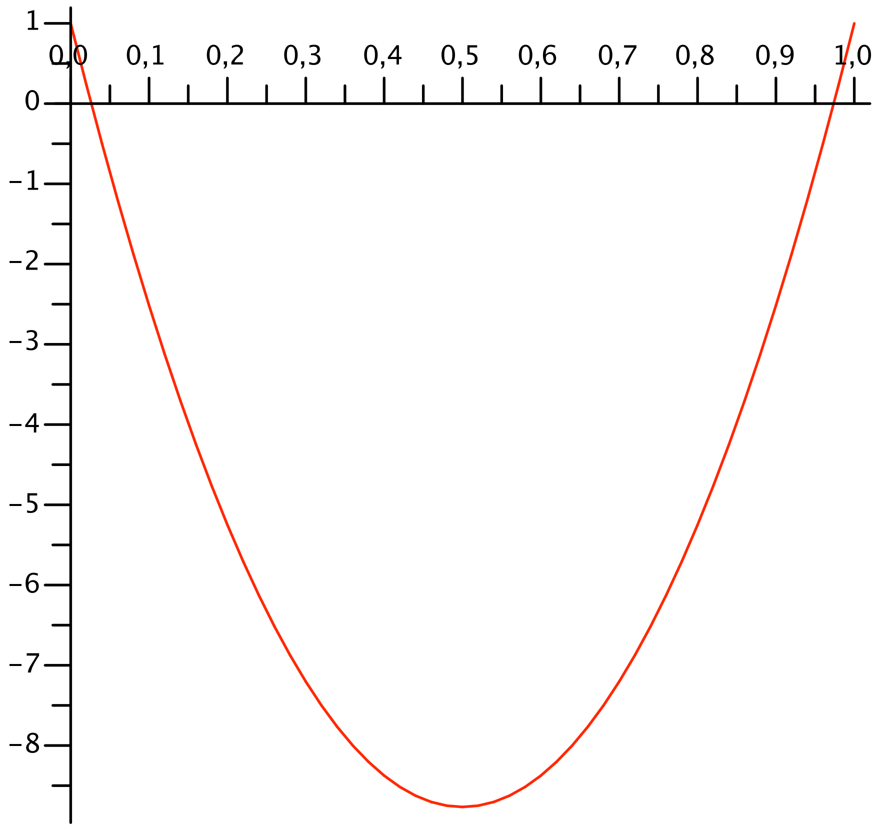
```
> Z2 := evalf(Newton_update(Z1));
```

$$Z2 := \begin{bmatrix} \text{1 .. 49 } Vector_{column} \\ \text{Data Type: } float_8 \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

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

$$Z3 := \begin{bmatrix} \text{1 .. 49 } Vector_{column} \\ \text{Data Type: } float_8 \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

```
> yyy := [1,seq(Z3[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
−1
−2
−3
−4
−5
−6
−7
−8
0,0
0,1
0,2
0,3
0,4
0,5
0,6
0,7
0,8
0,9
1,0

>