Soluzioni del compito di

Metodi Matematici e Calcolo per Ingegneria

del 12 luglio 2006

Enrico Bertolazzi

Trasformata di Laplace

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

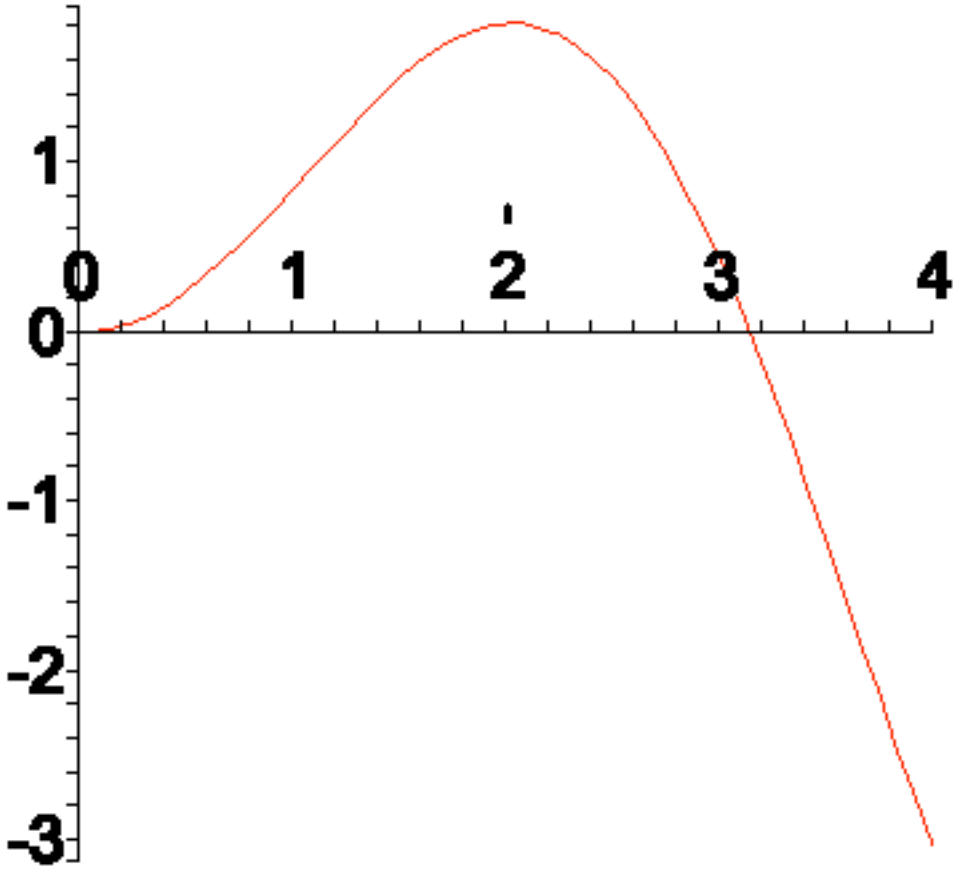
Data la seguente funzione

```
> #f := unapply( piecewise( t < Pi , sin(t), t >= Pi, 0 ) , t ) ;
```

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

$$f := t \to \sin(t)\, t$$

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



Usando le regole di trasformazione calcolare le trasformate delle funzioni

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

$$f1 := t \to f\left(\frac{1}{2}t\right)$$

$$f2 := t \to f\left(\frac{1}{3}t\right) \mathbf{e}^{(-3\,t)}$$

$$f3 := t \to \cos(t)\,t + \sin(t)$$

Trasformate con le primitive Maple

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

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

Calcolo le altre trasformate con le regole di trasformazione nelle tabelle

f(t/2) è un cambio di scala t --> t/2 diventa 2*F(2s)

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

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

Controllo con le primitive di Maple

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

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

f(t/3)*exp(-3*t) è un cambio di scala t --> t/3 per un esponenziale.
Spezzo la regola di trasformazione in vari passaggi:
- dalla regola del cambio di scala ho L(f(t/3)) = 3*F(3*s)
- definisco g(t) := f(t/3)
- definisco G(s)=3*F(3*s) la trasformata di g(t)
- la trasformata di g(t)*exp(-3*t) diventa G(s+3)
- quindi L(f(t/3)*exp(-3*t))=L(g(t)*exp(-3*t))=G(s+3)=3*F(3*(s+3))

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

$$\frac{18\,(s+3)}{(9\,s^2+54\,s+82)^2}$$

Controllo con le primitive di Maple

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

$$\frac{18\,(s+3)}{(9\,s^2+54\,s+82)^2}$$

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

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

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

Controllo con le primitive di Maple

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

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

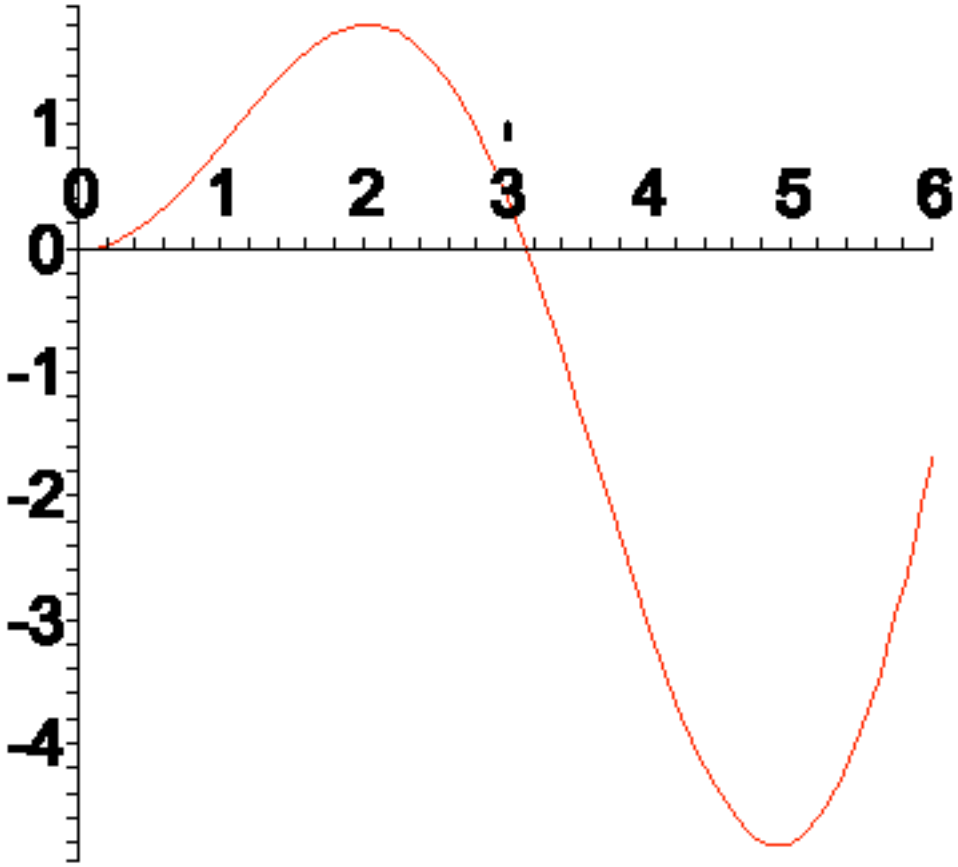
Soluzione di ODE con Laplace

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

```
> src := t -> sin(t)*t;
```

$$src := t \to \sin(t)\,t$$

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



Data la seguente equazione differenziale

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

$$ode := \frac{d}{dx} y(x) = \sin(x)\, x$$

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) ;
```

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

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

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

Risolvo la equazione per y(s)

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

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

Applico le condizioni iniziali ottenendo y(s)

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

$$ly := \frac{2}{(s^2+1)^2} + \frac{2}{s}$$

Espansione in fratti semplici

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

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

Antitrasformo per ottenere la equazione y(x)

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

$$res := \sin(t) - t\cos(t) + 2$$

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 := 2*yp     -zp = sin(t) ;
  ode2 :=  -yp +zp -wp = 0 ;
  ode3 :=      -zp +wp = 0 ;
```

$$ode1 := 2\left(\frac{d}{dt}y(t)\right) - \left(\frac{d}{dt}z(t)\right) = \sin(t)$$

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

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

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) ;
```

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

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

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

```
> 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);
```

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

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

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

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});
```

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

$$\left. laplace(z(t), t, s) = \frac{-1 + z(0)\,s^2 + z(0)}{s\,(s^2 + 1)} \right\}$$

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

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

$$SOL := \begin{bmatrix} \frac{1}{s} \\ \frac{1+2s^2}{s(s^2+1)} \\ \frac{s}{s^2+1} \end{bmatrix}$$

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) ;
```

$$yy := 1$$

$$zz := \cos(x) + 1$$

$$ww := \cos(x)$$

Soluzione di ricorrenza con trasformata zeta

```
> restart:
```

Risolvere la seguente ricorrenza

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

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

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));
```

$$-2k - (k+1)\left(\frac{1}{2}k+1\right)\left(\frac{1}{3}k+1\right) - 2 + 3(k+1)\left(\frac{1}{2}k+1\right)$$

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

$$\frac{2}{3}k - \frac{1}{6}k^3 + \frac{1}{2}k^2$$

Usando la Z-trasformata

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

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

Ricavo f(z)

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

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

Applico le condizioni iniziali

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

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

Conversione in fratti semplici

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

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

Inversione della Z-trasformata

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

$$f(k) = \frac{2}{3}\,k - \frac{1}{6}\,k^3 + \frac{1}{2}\,k^2$$

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 ;
```

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

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

Soluzione esatta

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

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

Matrice Jacobiano

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

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

Schema di Newton

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

$$Newton\_update := -\frac{x\,(y-1)}{5\,y^2+3\,x+y}\,e_x + \frac{y\,(3\,x+5\,y+1)}{5\,y^2+3\,x+y}\,e_y$$

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])) ;
```

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

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

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])) ;
```

$$x_1 := -0.04000000000$$

$$y_1 := 1.120000000$$

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])) ;
```

$$x_2 := 0.0006600660064$$

$$y_2 := 0.9980198020$$

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])) ;
```

$$x_3 := 2.185641840\ 10^{-7}$$

$$y_3 := 0.9999993443$$

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*z;
```

$$f := x\,y\,z$$

Soggetta ai vicoli

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

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

Soluzione con le primitive Maple

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

$$[0.187499999999999999,\ [x = 0.500000000000000000, y = 0.750000000000000000, z = 0.500000000000000000]]$$

Uso dei moltiplicatori di Lagrange

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

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

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

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

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

Sistema non lineare da risolvere

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

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

Soluzioni del sistema non lineare

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

$$_EnvExplicit := true$$

```
> RESunordered := [solve({seq(F[i],i=1..5)},{x,y,z,lambda,mu})] ;
```

$$RESunordered := \left[\left\{ z = \frac{1}{2}, \mu = \frac{1}{16}, x = \frac{1}{2}, y = \frac{3}{4}, \lambda = \frac{1}{4} \right\}, \left\{ x = \frac{1}{2} - \frac{1}{2} I, \mu = \frac{-1}{4}, z = \frac{1}{2} + \frac{1}{2} I, y = \frac{1}{2}, \lambda = \frac{1}{2} \right\}, \left\{ x = \frac{1}{2} + \frac{1}{2} I, \mu = \frac{-1}{4}, z = \frac{1}{2} - \frac{1}{2} I, y = \frac{1}{2}, \lambda = \frac{1}{2} \right.$$

]

Riordino le soluzioni

```
> SUBS := _x=x,_y=y,_z=z,_l=lambda,_m=mu ;
```

$$SUBS := \_x = x, \_y = y, \_z = z, \_l = \lambda, \_m = \mu$$

```
> RES :=
  [seq(sort(subs(RESunordered[i],[SUBS])),i=1..nops(RESunordered))];
```

$$RES := \left[\left[\_x = \frac{1}{2}, \_y = \frac{3}{4}, \_z = \frac{1}{2}, \_l = \frac{1}{4}, \_m = \frac{1}{16} \right], \left[\_x = \frac{1}{2} - \frac{1}{2} I, \_y = \frac{1}{2}, \_z = \frac{1}{2} + \frac{1}{2} I, \_l = \frac{1}{2}, \_m = \frac{-1}{4} \right], \left[\_y = \frac{1}{2}, \_l = \frac{1}{2}, \_m = \frac{-1}{4}, \_x = \frac{1}{2} + \frac{1}{2} I, \_z = \frac{1}{2} - \frac{1}{2} I \right] \right]$$

Prima soluzione

```
> S1 := subs(SUBS,RES[1]);
```

$$S1 := \left[x = \frac{1}{2}, y = \frac{3}{4}, z = \frac{1}{2}, \lambda = \frac{1}{4}, \mu = \frac{1}{16} \right]$$

Seconda soluzione (da scartare perchè è complessa)

```
> S2 := subs(SUBS,RES[2]);
```

$$S2 := \left[x = \frac{1}{2} - \frac{1}{2} I, y = \frac{1}{2}, z = \frac{1}{2} + \frac{1}{2} I, \lambda = \frac{1}{2}, \mu = \frac{-1}{4} \right]$$

Terza soluzione (da scartare perchè è complessa)

```
> S3 := subs(SUBS,RES[3]);
```

$$S3 := \left[y = \frac{1}{2}, \lambda = \frac{1}{2}, \mu = \frac{-1}{4}, x = \frac{1}{2} + \frac{1}{2} I, z = \frac{1}{2} - \frac{1}{2} I \right]$$

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 ;
```

$$\begin{bmatrix} 0 & z & y \\ z & 0 & x \\ y & x & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

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

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

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

Controllo minimo/massimo locale primo punto

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

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

$$\mu 1 := \frac{1}{16}$$

Calcolo l'Hessiano nel punto stazionario

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

$$Hf1 := \begin{bmatrix} 0 & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{4} & 0 & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{4} & 0 \end{bmatrix}$$

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

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

$$\begin{bmatrix} -0.5000000000 \\ 0.6830127020 \\ -0.1830127020 \end{bmatrix}$$

Cerco nello spazio dei vincoli:

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

$$-\alpha^2$$

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

$$\frac{3}{16}$$

Approssimazione di un problema del calcolo delle variazioni

```
> restart:
```

Integrale da minimizzare

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

$$\int_0^1 y(x)\left(\frac{d}{dx}y(x)\right)^2 dx$$

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) ;
```

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

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

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

```
> 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));
```

$$Z2 := 0.9512040199000000030\, e_x + 0.934561965500000035\, e_y + 0.9512040199000000030\, e_z$$

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

$$Z3 := 0.9512004214000000051\, e_x + 0.934555355300000001\, e_y + 0.9512004214000000051\, e_z$$

Verica del residuo di GRAD F

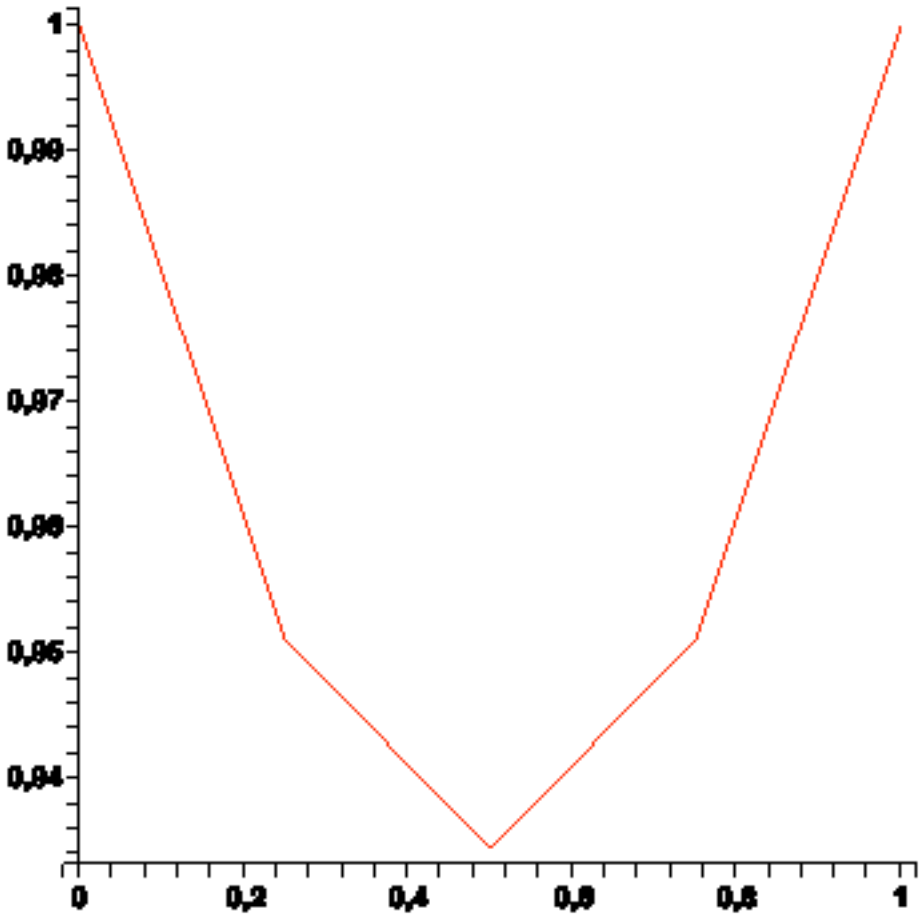
```
> subs(seq(y[i]=Z3[i],i=1..n-1),y[0]=ya,y[n]=yb,eqns) ;
```

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

```
> 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)));
```

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

```
> 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));
```

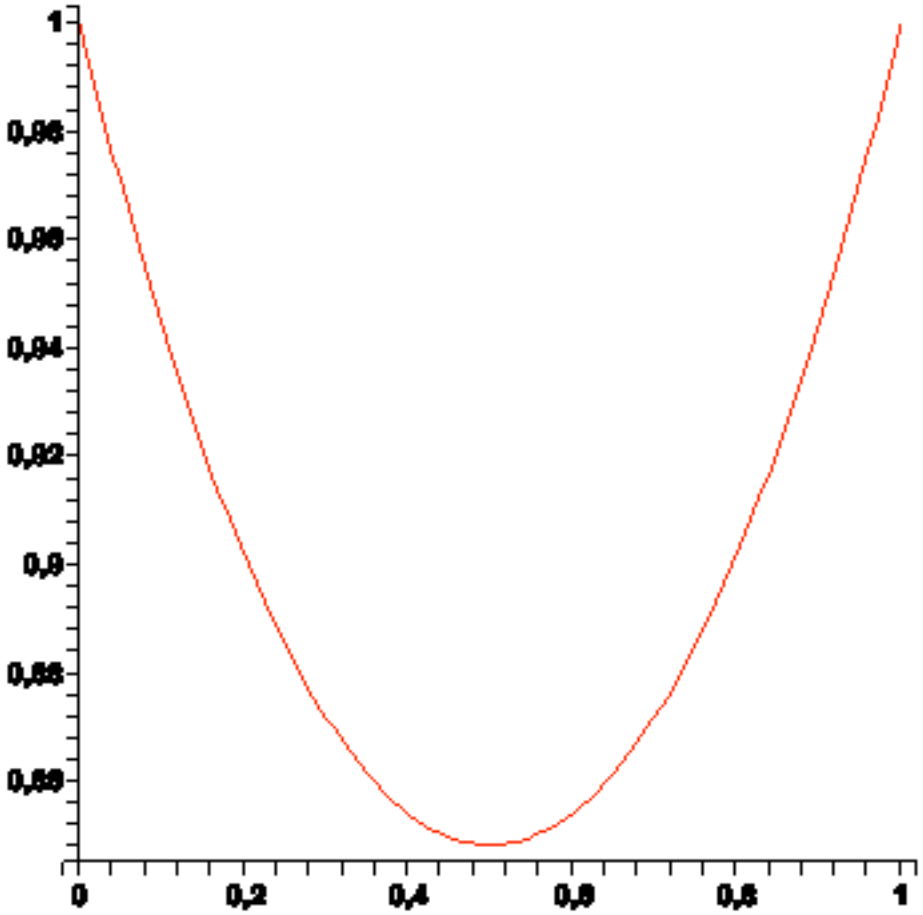
$$Z6 := \begin{bmatrix} 1 .. 99\ Vector[column] \\ Data\ Type: float[8] \\ Storage: rectangular \\ Order: Fortran\_order \end{bmatrix}$$

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);
```

0.00559964299999999876

```
> 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
0,96
0,98
0,94
0,92
0,9
0,88
0,86
0
0,2
0,4
0,6
0,8
1

>