Elementi di Informatica e Applicazioni Numeriche T

Controllo degli Errori
(Accenni)

Un Esempio

Un serbatoio a pressione atmosferica contiene un reagente chimico

- Il reagente ha $\rho = 1.189 \times 10^3$ e $\mu = 2.10 \times 10^{-3}$
- Il livello del serbatoio è di $2\ m$ di reagente

Il serbatoio scarica il reagente attraverso una condotta

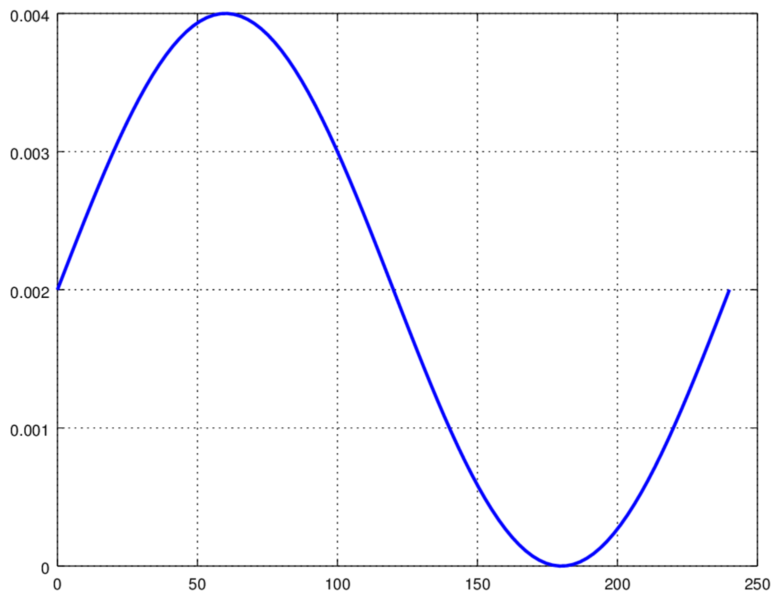
- La condotta è lunga $30\ m$ e scende di $1\ m$ complessivamente
- La condotta ha un diametro di $5\ cm$ e scabrezza $2\ mm$

Il serbatoio viene riempito di reagente nel tempo, secondo la legge:

$$Q_{in} = 0.002\left(1 + \sin\left(2\pi\frac{t}{240}\right)\right)$$

Un Esempio

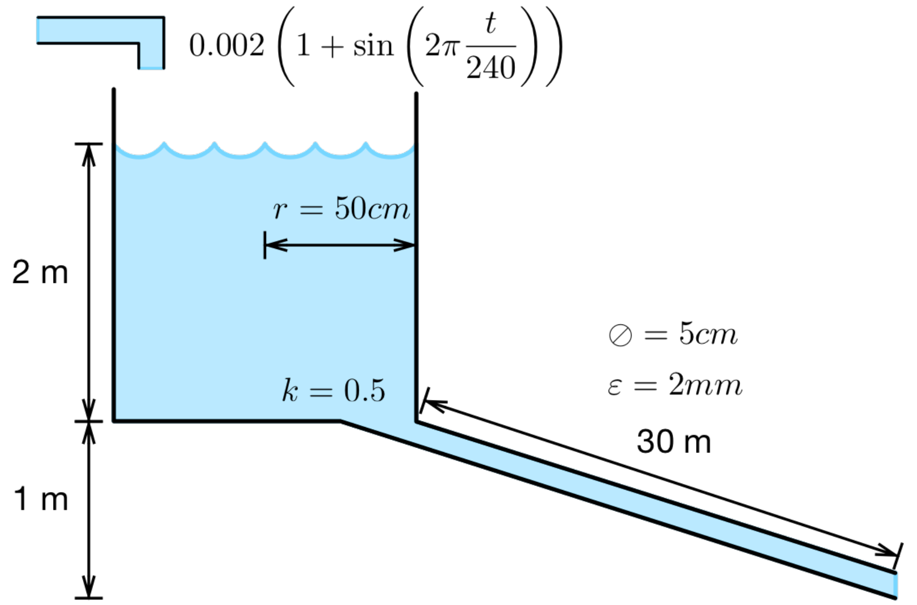
In pratica, la portata in ingresso varia con un periodo di $240\ s$



- La portata massima in ingresso è di $2\ L/s$
- La portata mininima è nulla

Un Esempio

I dati del problema in sintesi:



- Si determini il livello del serbatoio nel tempo

Soluzione

Si tratta di un sistema dinamico tempo continuo

- Quindi possiamo modellarlo con una EDO…
- …E per prime cosa dobbiamo scegliere come modellare lo stato

Una possibilità: $H =$ volume del reagente

- Con questa scelta, la derivata è data da:

$$dH = Q_{in} - Q_{out}$$

- Dove Q_{out} è la portata in uscita lungo la condotta

Soluzione

Per calcolare Q_{out} , codifichiamo l'equazione di Bernoulli...

```
function z = bernoulli(Q, h1)
    ...
    v = hspeed(Q, D);
    H12 = z1 + h1 - v.^2 ./ (2.*g);
    nu = mu ./ rho;
    Re = Reynolds(v, D, nu);
    f = churchill(Re, es ./ D);
    WD = loss_l(f, D, L, v, g);
    WC = loss_k(k, v, g);
    z = H12 - WD - WC;
end
```

- Il codice intero è disponibile sul sito del corso

Soluzione

...Poi determiniamo il valore della portata con il metodo di Newton:

```
function Q = Qout_computation(h1)
    % Definisco una funzione da usare con fsolve
    f = @(Q) ( bernoulli(Q, h1) );
    % Determino la portata
    Q = fsolve(f, 0.001);
end
```

- **ATTENZIONE:** non è detto che il metodo di Newton converga!
- Quando possibile, è bene controllare che non ci siano errori...

Controllo degli Errori in Octave

In Octave possiamo indicare la presenza di errori con:

```
error(MSG)
```

- `MSG` è una stringa (un messaggio di errore)

La funzione:

- Interrompe l'esecuzione
- Stampa il messaggio di errore
- Indica la riga in cui `error` è stata chiamata

Soluzione

Nel nostro caso, possiamo controllare il risultato con:

```
function Q = Qout_computation(h1, tol = 1e-5)
    % Definisco la funzione da usare con fsolve
    f = @(Q) ( bernoulli(Q, h1) );
    % Determino la portata
    [Q, FVAL] = fsolve(f, 0.001);
    if abs(FVAL) > tol % Vero in caso di errori
        error('Errore nel calcolo della portata')
    end
end
```

- Se `fsolve` ha prolemi a convergere (e.g. `h1` negativo)...
- ...Il nuovo codice segnala che c'è stato un problema!

Soluzione

Adesso che abbiamo Q_{out} , possiamo calcolare dH

```
function dH = dH_computation(H, t)
    ...
    % Calcolo l'altezza del serbatoio
    h1 = H ./ (pi .* r.^2);
    % Calcolo Qout
    Qout = Qout_computation(h1);
    % Calcolo Qin
    Qin = 0.002.*(1 + sin(2.* pi .* (t ./ 240)));
    % Calcolo la derivata del volume di reagente
    dH = -Qout + Qin;
end
```

Soluzione

Sempre per evitare errori, gestiamo il caso $h1 = 0$

```
function dH = dH_computation(H, t, tol = 1e-5)
    ...
    Qout = 0;
    if h1 > tol
        Qout = Qout_computation(h1);
    end
    ...
end
```

- Se non lo facciamo ed il serbatoio si svuota...
- ...l'equazione di Bernoulli fornisce ancora una portata non nulla!
- Questo succede perché il serbatoio è rialzato

Soluzione

Per terminare l'esercizio dobbiamo solo risolvere una EDO

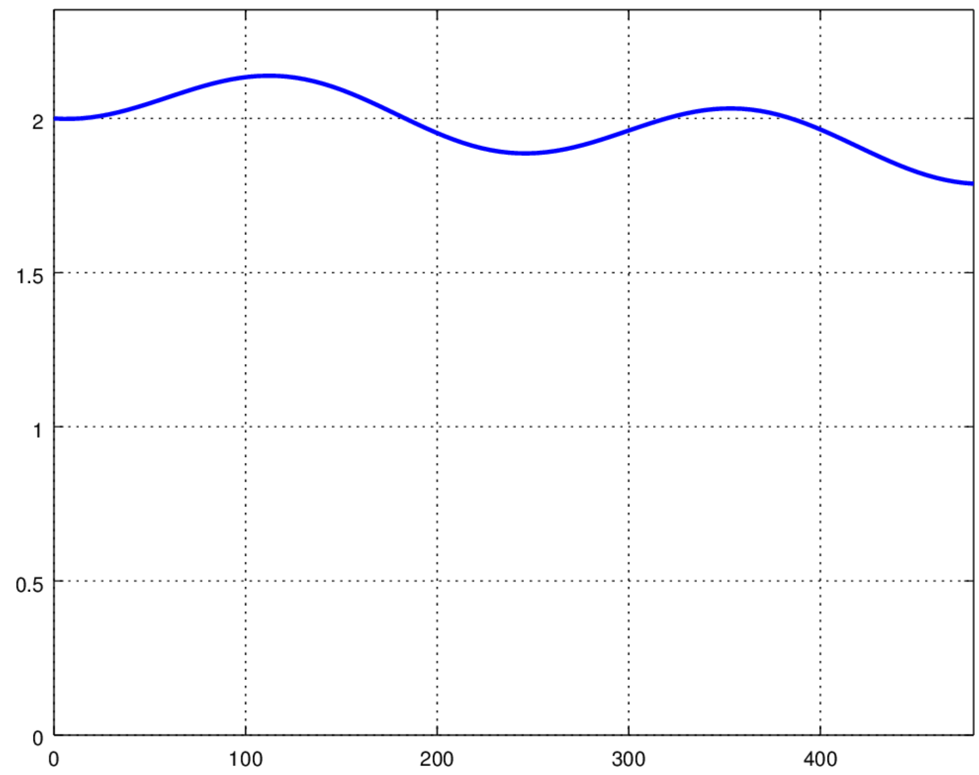
```
t = linspace(0, 480); % Intervallo di tempo
H0 = 2 .* (pi .* r.^2); % Volume iniziale
H = lsode(@dH_computation, H0, t);

% Trasformo il volume in livello
Hl = H ./ (pi .* r.^2);

% Disegno l'andamento
plot(t, Hl)
```

Soluzione

Ecco l'andamento per i primo 480 secondi:

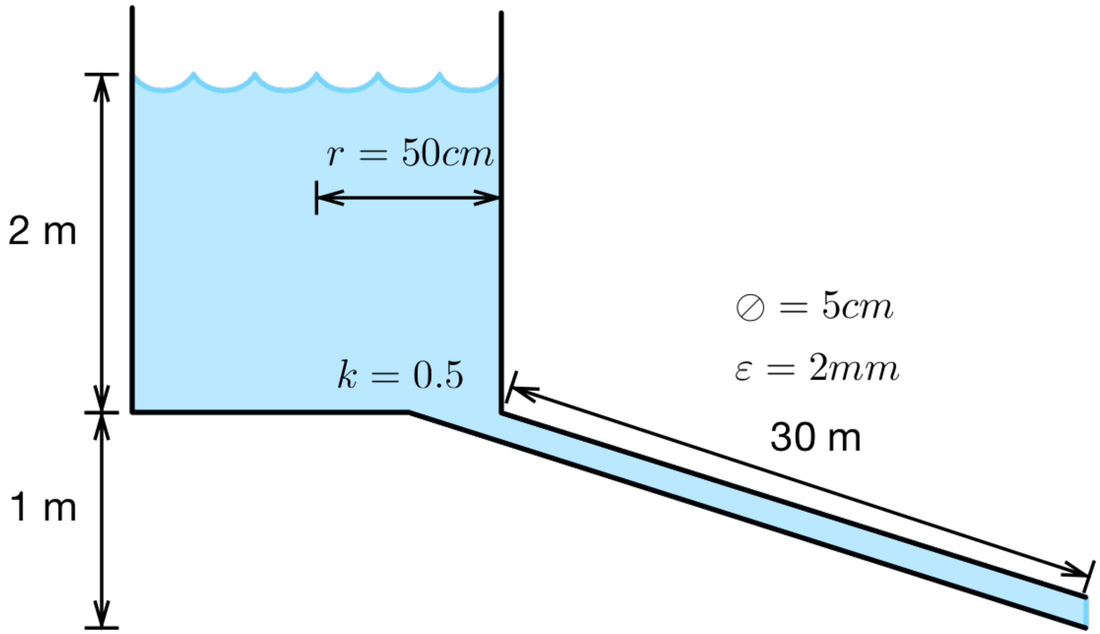


Elementi di Informatica e Applicazioni Numeriche T

Inversione del Tempo
e Stato Stazionario

Un Esempio

Supponiamo che il serbatoio dell'esercizio visto non sia alimentato



- Quale deve essere il livello iniziale perché si svuoti in $300\ s$?

Soluzione

È sufficiente invertire la direzione della variabile tempo

Quindi dobbiamo negare le derivate:

```
dH = +Q; % Era: dH = -Q (+ Qin)
```

Poi sostituire le occorrenze di t con $t_f - t$

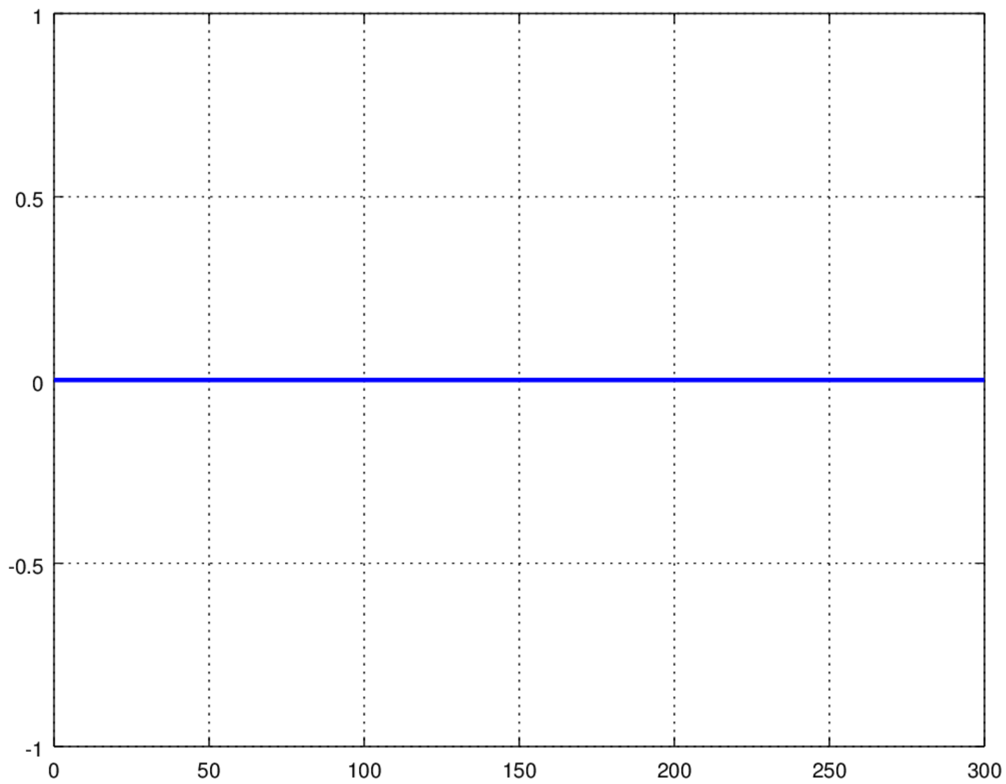
- Nel nostro caso **`dH`** non dipende dal tempo

Quindi dobbiamo risolvere l'EDO:

```
t = linspace(0, 300);
H0 = 0;
H = lsode(@dH_computation, H0, t);
```

Soluzione

Se lo facciamo, otteniamo questo risultato:



Inversione del Tempo e Stato Stazionario

Perché il livello è piatto?

Perché siamo partiti da uno stato stazionario valido!

- Allo stato stazionario la derivata è nulla

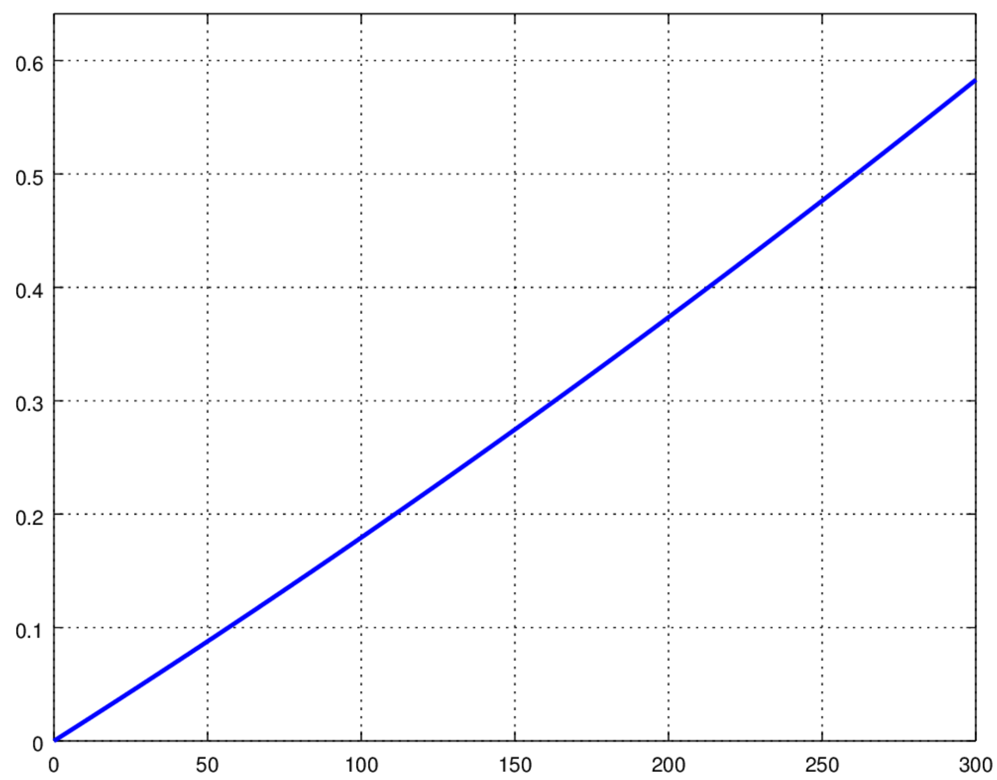
Soluzione:

- Basta partire da un serbatoio "leggermente pieno":

```
t = linspace(0, 300);
H0 = tol; % Es. tol = 1e-4
H = lsode(@dH_computation, H0, t);
```

Soluzione

Ecco la nuova soluzione:



Elementi di Informatica e Applicazioni Numeriche T

Equazioni Non Lineari
e Problemi di Progettazione

Un Esempio

- Consideriamo il serbatoio del caso precedente...
- ...Ma assumiamo che il serbatoio non sia alimentato

Supponiamo che il reagente sia pericoloso:

- La condotta è normalmente chiusa (alla bocca)...
- ...In caso di problemi, viene aperta e deve svuotare il serbatoio
- HP: il serbatoio si considera vuoto anche se la condotta non lo è

Problema:

- Quale deve essere il diametro della condotta...
- ...Per svuotare il serbatoio in 60 secondi?

Problemi di Progettazione

Questo tipo di problema si può ridurre ad una equazione

Supponiamo di avere una funzione $F(D)$ che:

- Dato un valore per il diametro D...
- ...Denota il livello del serbatoio dopo $60\ s$

Allora, il nostro problema si riduce a risolvere:

$$F(D) = 0$$

- $F(D)$ non può essere espressa con una formula analitica...
- ...Ma non importa! Ci basta saperla calcolare

Soluzione

La nostra funzione dovrà avere più o meno questa struttura:

```
function H60 = H_in_60_sec(D)
    ...
    t = linspace(0, 60); % 60 secondi bastano
    H0 = 2 .* (pi .* r.^2);
    f = @(H, t) (...); % Funzione per calcolare dH
    H = lsode(f, H0, t);
    % Restituisco il volume dopo 60 secondi
    H60 = H(end);
end
```

- Internamente, risolviamo una equazione differenziale!
- È semplicemente il nostro modo per calcolare il livello dopo $60\ m$
- Dobbiamo capire come calcolare `dH`, però...

Soluzione

Per calcolare `dH`, modifichiamo la funzione dell'esercizio 1:

```
function dH = dH_computation(H, t, D)
    ...
    h1 = H ./ (pi .* r.^2);
    Qout = ... % Qout dipende anche da D!
    Qin = 0.002.*(1 + sin(2.* pi .* (t ./ 240)));
    dH = Qout + Qin;
end
```

- E nella funzione `H_in_60_sec`, avremo:

```
f = @(H, t) dH_computation(H, t, D);
```

- Adesso dobbiamo solo poter calcolare `Qout`...

Soluzione

Per calcolare **`Qout`**, modifichiamo la funzione dell'esercizio 1:

```
function Q = Qout_computation(h1, D, tol=1e-5)
    f = @(Q) ( ... ); % Questa dipenderà da D!
    [Q, FVAL] = fsolve(f, 0.001);
    if abs(FVAL) > tol
        error('Errore nel calcolo della portata')
    end
end
```

- E nella funzione **`dH_computation`**, avremo:

```
Qout = Qout_computation(h1, D);
```

- A questo punto ci manca solo l'equazione di Bernoulli...

Soluzione

Ora dobbiamo solo modificare un'ultima funzione dell'esercizio 1:

```
function z = bernoulli(Q, h1, D)
    % Il codice è invariato!
end
```

- Nella funzione `Qout_computation`, avremo:

```
f = @(Q) bernoulli(Q, h1, D);
```

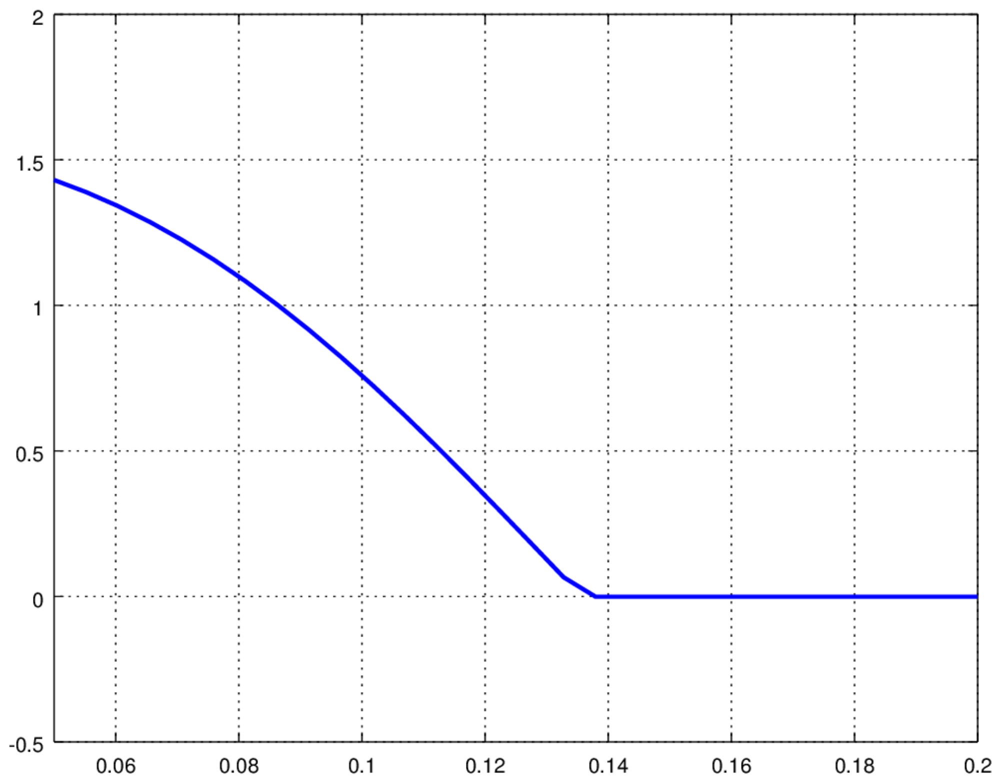
A questo punto possiamo risolvere il problema con:

```
D = fsolve(H_in_60_sec, 0.05)
```

- Ma non funziona. Perché?

Soluzione

Vediamo come varia il livello a $60\ s$ in funzione di D



Soluzione

- Il metodo di Newton non può procedere...
- ...Se visita un valore di D dove $F'(D) = 0$

Nel nostro caso ci sono molti punti in cui $F(D)$ è "piatta"

Soluzione:

- Facciamo sì che quando il livello è 0, **`H60`** sia leggermente negativo:

```
H60 = H(end) - 1e-5;
```

- E poi usiamo un metodo "tipo bisezione"!

```
D = fzero(H_in_60_sec, [0.05, 0.30])
```

Elementi di Informatica e Applicazioni Numeriche T

Interpolazione con Vincoli

Un Esempio

Nostro nipote ci ha chiesto di costruire una pista per le macchinine:

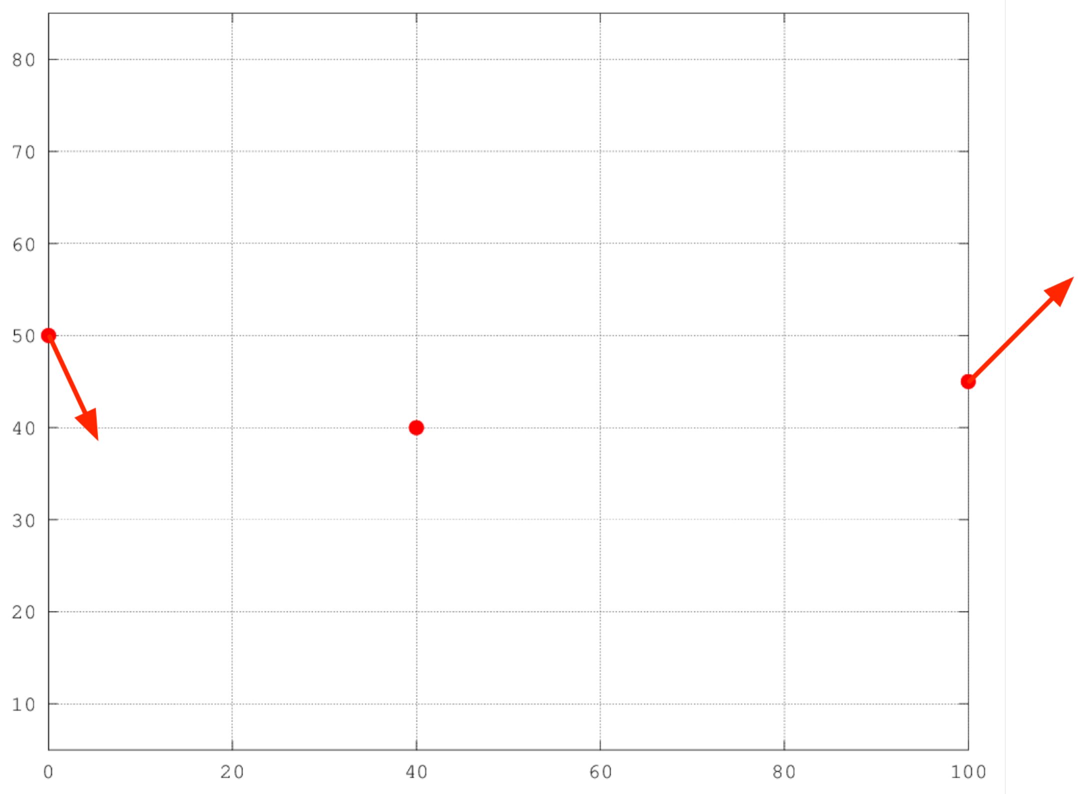
- La pista deve essere lineare, ma può andare in salita e discesa
- La quota iniziale deve essere di $50\ cm$
- Dopo $40\ cm$, la quota deve essere di $40\ cm$
- La pista termina dopo $1\ m$ ad una quota di $45\ cm$
- La pendenza iniziale deve essere di $-2\ cm$ per cm
- La pendenza finale deve essere di $+1\ cm$ per cm

Determinare la forma della pista utilizzando una curva polinomiale

- Evidentemente si tratta di un nipotino esigente :-)

Un Esempio

Ecco i dati del problema in sintesi:



Problemi di Interpolazione Vincolati

- Si tratta di un problema di interpolazione "vincolato"...
- ...E si risolve più o meno nello stesso modo!

Proviamo a formalizzare cosa sappiamo del problema:

- Diamo un nome ai nostri punti di riferimento:

$$x_0 = 0 \qquad x_0 = 40 \qquad x_2 = 100$$
$$y_0 = 50 \qquad y_1 = 40 \qquad y_2 = 45$$

- Sappiamo che la curva deve essere polinomiale:

$$f(x, \beta) = \sum_{j=0}^{n} \beta_j x^j$$

Problemi di Interpolazione Vincolati

A questo punto possiamo formulare il problema:

$$
\begin{aligned}
f(x_0, \beta) &= y_0 \\
f(x_1, \beta) &= y_1 \\
f(x_2, \beta) &= y_2 \\
f'(x_0, \beta) &= -1 \\
f'(x_1, \beta) &= +1
\end{aligned}
$$

- Le condizioni sulla pendenza sono condizioni sulla derivata

È un sistema di equazioni! Le incognite sono i parametri β

- Ci sono 5 equazioni, quindi ci devono essere 5 incognite
- Conseguenza: la curva è un polinomio di grado 4

Soluzione

Possiamo risolvere il sistema con il metodo di Newton:

```
function z = f(p)
    p1=p(1); p2=p(2); p3=p(3); p4=p(4); p5=p(5);
    z(1)= p1*x0^4 +p2*x0^3 +p3*x0^2 +p4*x0 +p(5) -y0;
    z(2)= p1*x1^4 +p2*x1^3 +p3*x1^2 +p4*x1 +p(5) -y1;
    z(3)= p1*x2^4 +p2*x2^3 +p3*x2^2 +p4*x2 +p(5) -y2;
    z(4)= 4*p1*x0^3 +3*p2*x0^2 +2*p3*x0 +p4 -(-1);
    z(5)= 4*p1*x2^3 +3*p2*x2^2 +2*p3*x2 +p4 -(+1);
end

p = fsolve(@f, [0, 0, 0, 0, 0])
```

- `p` corrisponde a β nelle nostre equazioni

Soluzione

Il sistema è lineare in β (risp. `p`), quindi una alternativa è:

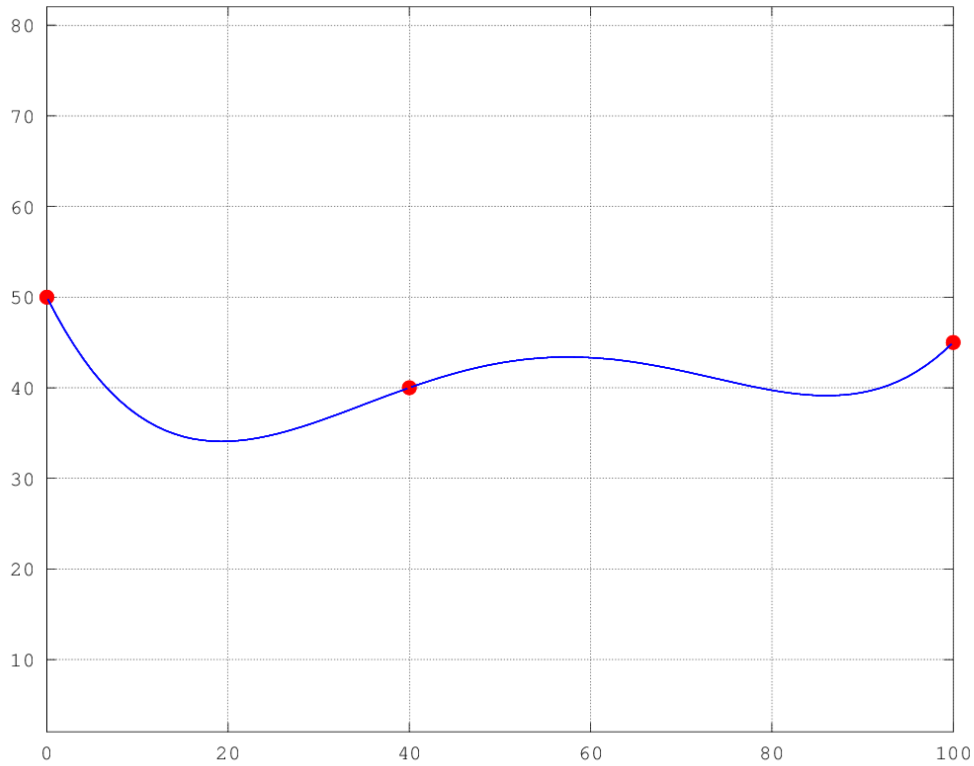
```
A = [
    x0.^4 , x0.^3 , x0.^2 , x0 , 1;
    x1.^4 , x1.^3 , x1.^2 , x1 , 1;
    x2.^4 , x2.^3 , x2.^2 , x2 , 1;
    x0.^3 , x0.^2 , x0    , 1  , 0;
    x2.^3 , x2.^2 , x2    , 1  , 0;
];

p = A \ [y0; y1; y2; 1; -1];
```

- Il risultato è lo stesso di prima

Soluzione

La nostra pista:



Elementi di Informatica e Applicazioni Numeriche T

Massimi e Minimi Non Vincolati

Massimi e Minimi Non Vincolati

Consideriamo il problema seguente:

- Data la pista costruita nell'esercizio precedente…
- …Si determini la quota minima e la quota massima

Si tratta di un problema di ottimizzazione

Vogliamo determinare il massimo/minimo

- Di su una funzione univariata (singola variabile)
- Senza vincoli addizionali sul massimo/minimo

Quindi si può risolvere con metodi analitici classici

Massimi e Minimi Non Vincolati

In particolare, possiamo:

- Calcolare la derivata della funzione che descrive la pista:
- Risolvere l'equazione:

$$f'(x) = 0$$

Le soluzioni rappresentano i punti stazionari

- Se la funzione è univariata, corrispondono a massimi o minimi
- Per distinguere massimi e minimi possiamo guardare il plot
- Oppure possiamo calcolare la derivata seconda:

$$f''(x_0) > 0 \leftrightarrow \text{minimo} \qquad f''(x_0) < 0 \leftrightarrow \text{massimo}$$

Soluzione

Vediamo come procedere nel nostro caso:

La funzione è polinomiale, quindi facile da derivare:

```
pp = polyder(p);
fp = @(x) polyval(pp, x);
```

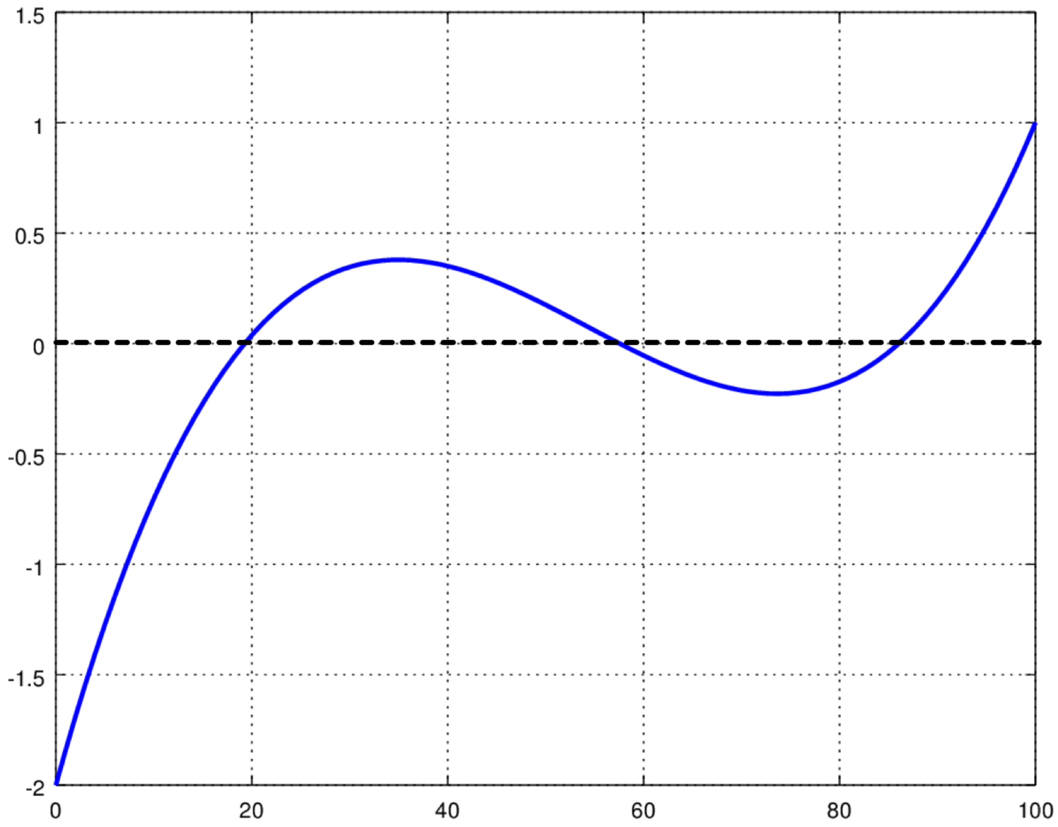
Per trovare i punti stazionari potremmo usare:

```
x_sol = fsolve(fp, xa)
x_sol = fsolve(fp, [xa, xb])
```

- Agendo su `xa` (per `fsolve`) o su `xa` e `xb` (per `fzero`)...
- ...Possiamo controllare quale soluzione venga restituita

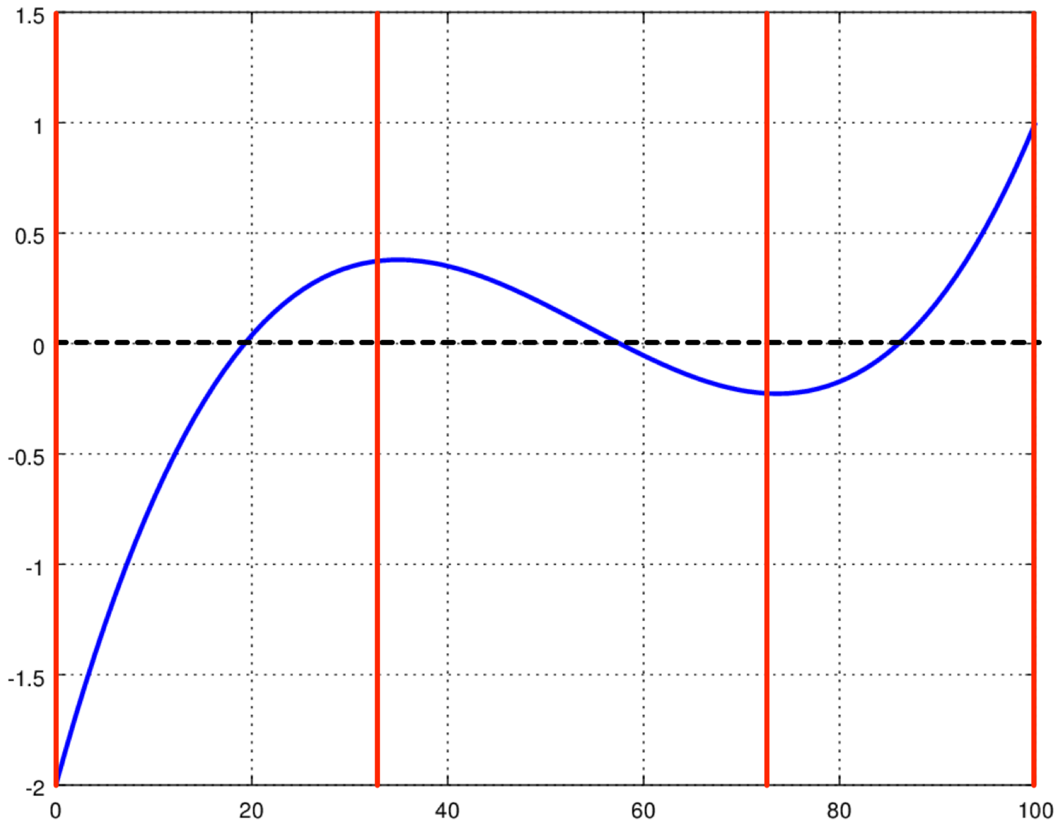
Soluzione

Partiamo dal grafico della derivata della nostra curva $f_\beta(x)$:



Soluzione

Su di esso possiamo individuare tre zone che racchiudono gli zeri:



Soluzione

Possiamo sfruttare le zone per trovare gli zeri con **`fzero`**:

```
f = @(x) ( polyval(p, x) );
fp = @(x) ( polyval(pp, x) );

m0x = fzero(fp, [0, 30])
m0y = f(m0x) % Altezza della pista

m1x = fzero(fp, [30, 70])
m1y = f(m0x) % Altezza della pista

m2x = fzero(fp, [70, 100])
m2y = f(m0x) % Altezza della pista
```

- Si può anche usare **`fsolve`**, partendo da una punto vicino allo 0

Altri Problemi di Ottimizzazione (Accenni)

- I problemi di ottimizzazione sono molto importanti in pratica
- In questo esempio abbiamo considerato un caso di:

Ottimizzazione univariata, non vincolata

$$\min_{x \in \mathbb{R}} z = f(x)$$

- Si può ridurre ad una equazione nella forma $f'(x) = 0$
- Se ci sono molti punti stazionari, <u>vanno esplorati tutti</u>
- Ci può volere molto tempo (nel caso peggiore: secoli!)...
- ...Quindi a volte ci si ferma prima (non si ottiene il "vero" minimo)

Ci sono alti casi interessanti (li accenniamo soltanto)...

Altri Problemi di Ottimizzazione (Accenni)

Ottimizzazione multivariata, non vincolata

- E.g. Modificare i parametri di controllo di un reattore…
- …Per ridurre al minimo la velocità di reazione

Ottimizzazione multivariata, vincolata

- E.g. Garantire una certa portata su un sistema idrico…
- …E minimizzarne il costo di costruzione

Ottimizzazione multivariata, vincolata, misto-discreta

- E.g. Progettare le traiettorie degli autobus di Bologna…
- …Per massimizzare la qualità del servizio

Elementi di Informatica e Applicazioni Numeriche T

Lunghezza di una Curva

Lunghezza di una Curva

Si determini la lunghezza della pista dell'esercizio 3

- Intuitivamente, si tratta di un problema di integrazione
- Ma cosa dobbiamo integrare?

Approssimiamo la curva con tanti segmenti lineari

- Ogni segmento avrà lunghezza infinitesima
- E sarà orientato secondo le tangente

La tangente in un punto x è definita dall'equazione:

$$y = f(x) + f'(x)(x' - x)$$

Lunghezza di una Curva

Quindi, rispetto ad un punto di riferimento x:

- Un segmento sulla tangente corrisponde ad un vettore:

$$\begin{pmatrix} \Delta x \\ f'(x)\Delta x \end{pmatrix}$$

- Ed ha lunghezza:

$$\sqrt{(\Delta x)^2 + (\Delta x)^2 f'(x)^2} = \Delta x \sqrt{1 + f'(x)^2}$$

- Se lo spostamento è infinitesimo, abbiamo:

$$dx\sqrt{1 + f'(x)^2}$$

Lunghezza di una Curva

Quindi per ottenere la lunghezza, ci basta calcolare:

$$L_{a,b} = \int_a^b \sqrt{1 + f'(x)^2} dx$$

Nel nostro esempio:

```
p = % coefficienti del polinomio
pp = polyder(p);
phip = @(x) polyval(pp, x); % derivata della curva

% Definisco la funzione da integrare
ddist = @(x) sqrt(1 + phip(x).^2);
% Integro tra i due estremi
L = quadv(ddist, x0, x2)
```