# Equazioni differenziali con valori al bordo

Lucia Gastaldi

DICATAM - Sez. di Matematica,
http://www.ing.unibs.it/gastaldi/

# Indice

# Equazioni differenziali con valori ai limiti

Si consideri l'equazione differenziale

$$\begin{array}{l} -\mu u''(x) + \sigma(x) u(x) = f(x) \quad \text{per } x \in (a, b) \\ u(a) = \alpha, \quad u(b) = \beta. \end{array}$$

Suddividiamo l'intervallo $[a, b]$ in $N + 1$ parti uguali e poniamo $h = (b - a)/(N + 1)$. Poniamo poi $x_i = a + ih$.
Possiamo approssimare la derivata seconda con la seguente differenza finita del secondo ordine

$$u''(x_i) \approx \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} \quad \text{per } i = 1, \ldots, N.$$

Indicando con $u_i^h$ il valore approssimato di $u(x_i)$, si ottiene il seguente sistema lineare:

$$\begin{array}{l} -\mu \dfrac{u_{i+1}^h - 2u_i^h + u_{i-1}^h}{h^2} + \sigma(x_i) u_i^h = f(x_i) \quad \text{per } i = 1, \ldots, N \\ u_0^h = \alpha \qquad u_{N+1}^h = \beta \end{array}$$

# Equazioni differenziali con valori ai limiti II

Quindi si ricava la soluzione $u_i^h$ per $i = 1, \ldots, N$ risolvendo il sistema

$$A_h u^h = b$$

essendo $A_h$ la matrice tridiagonale che ha i seguenti elementi

$$a_{ii} = 2\mu/h^2 + \sigma(x_i), \quad a_{ii-1} = a_{ii+1} = -\mu/h^2$$

e il termine noto è dato da

$$\begin{aligned} b_1 &= f(x_1) + \mu\alpha/h^2, \\ b_i &= f(x_i) \text{ per } i = 2, \ldots, N-1, \\ b_N &= f(x_N) + \mu\beta/h^2. \end{aligned}$$

## Stima dell'errore

$$\max_{1 \le i \le N} |u(x_i) - u_i^h| \le Ch^2 \max_{a \le x \le b} |u^{(4)}(x)|$$

# Costruzione della matrice

Per costruire la matrice del sistema lineare che si ottiene con le differenze finite osserviamo che si può scrivere $A = (\mu/h^2)K + M$ essendo

$$K = \begin{bmatrix} 2 & -1 & \dots & \dots & \dots & \dots & 0 \\ -1 & 2 & -1 & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & -1 & 2 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & -1 & 2 & -1 \\ 0 & \dots & \dots & \dots & \dots & -1 & 2 \end{bmatrix}$$

$$M = \begin{bmatrix} \sigma(x_1) & 0 & \dots & \dots & 0 \\ 0 & \sigma(x_2) & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & \sigma(x_N) \end{bmatrix}$$

# Costruzione del termine noto

Il termine noto può essere scritto come la somma di due vettori $b = F + bc$: $F$ tiene conto del dato $f$ sull'intervallo, mentre $bc$ è relativo alle condizioni al bordo.

$$F = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \dots \\ \dots \\ f(x_N) \end{bmatrix} \qquad bc = \begin{bmatrix} \mu\alpha/h^2 \\ 0 \\ \dots \\ 0 \\ \mu\beta/h^2 \end{bmatrix}$$

# Soluzione dell'equazione differenziale

Scrivere la function eqlim per calcolare la soluzione dell'equazione differenziale con valori ai limiti

$$\begin{array}{l} -\mu u''(x) + \sigma(x)u(x) = f(x) \quad \text{per } x \in [a,b] \\ u(a) = \alpha, \quad u(b) = \beta. \end{array}$$

mediante il seguente comando:

```
[x,u]=eqlim(f,sigma,mu,a,b,alfa,beta,N)
```

**Input**

| | |
|---|---|
| `f,sigma` | nome delle function_handle che contengono l'espressione analitica di $f$ e $\sigma$ |
| `mu` | coefficiente davanti alla derivata seconda |
| `a,b` | estremi dell'intervallo |
| `alfa,beta` | valori ai limiti |
| `N` | numero di punti in cui si calcola la soluzione |

**Output**

| | |
|---|---|
| `x` | ascisse dei punti in cui si calcola la soluzione |
| `u` | valori della soluzione |

# Function `eqlim`

La function `eqlim` si compone di 5 passi:

- assegnazione della griglia di calcolo: calcolo $h$ e i punti $x_i = a + ih$ per $i = 1, \ldots, N$;
- costruzione della matrice $A_h$ come suggerito usando i comandi `ones` e `diag`;
- costruzione del termine noto (attenzione al nome);
- risoluzione del sistema lineare;
- organizzazione dell'output che tenga conto delle condizioni al bordo.

Oss Si possono completare i vettori `x` e `u` con i valori negli estremi dell'intervallo. Per fare ciò si devono aggiungere le rispettive componenti all'inizio ed alla fine del vettore, con i comandi

- `x=[a,x,b]` se `x` è un vettore riga;
- `u=[alfa;u;beta]` se `u` è un vettore colonna.

# Esercizio 1

Risolvere l'equazione differenziale

$$
\begin{aligned}
&-u''(x) + \sin x \, u(x) = (1 - \cos^2 x - 2\cos x)e^x \quad \text{per } x \in [0, \pi] \\
&u(0) = u(\pi) = 0.
\end{aligned}
$$

Confrontare i risultati ottenuti con la soluzione esatta
$u(x) = \sin x \, e^x$.

Scrivere un file di tipo script che:

- assegna i dati, in particolare le funzioni $f$ e $\sigma$ di tipo `@`;
- calcola la soluzione usando la function `eqlim`;
- plotta la soluzione discreta insieme a quella continua;
- calcola l'errore

$$E = \max_{1 \le i \le N} |u(x_i) - u_i^h|.$$

Errore Per calcolare l'errore usare il comando `norm` (si veda la sezione alla fine delle slides).

# Convergenza del metodo delle differenze finite

### Esercizio 2

Dato `N=[10 20 40 80 160 320]`, per ciascun valore di $N$

- calcolare la soluzione dell'equazione differenziale dell'esercizio precedente;
- riportare la soluzione in un grafico insieme alla soluzione esatta;
- valutare l'errore relativo.

Alla fine riportare gli errori in funzione di `N` in un grafico in scala bilogaritmica insieme all'andamento teorico.
Calcolare l'ordine di convergenza del metodo.

# Convergenza del metodo delle differenze finite

## Svolgimento dell'Esercizio 2

Scrivere un file di tipo script per eseguire i seguenti passi.

- Assegnare i dati: $f$, $\sigma$, $a$, $b$, $\alpha$, $\beta$ e la soluzione esatta *sol*.
- Assegnare `N=[10 20 40 80 160 320]`.
- Usando un ciclo `for` per ciascun valore di $N$:
  - calcolare la soluzione `u`, usando la function `eqlim`;
  - valutare la soluzione esatta nei punti della griglia;
  - riportare in uno stesso grafico la soluzione esatta e quella discreta;
  - calcolare l'errore con il comando `E(i)=norm(u-sol,inf)` (`i` rappresenta la varibile del ciclo `for`).
- Riportare gli errori in funzione di `N` in un grafico in scala bilogaritmica: `loglog(N,E)`.
- Calcolare l'ordine del metodo con il comando:

```
p=(log(E(2:end))-log(E(1:end-1)))./...
                (log(N(1:end-1))-log(N(2:end)))
```

# Esercizio 3

Dato $q \geq 2$, si consideri l'equazione differenziale

$$-u''(x) = q(q-1)|x|^{q-2} \text{ per } x \in (-1,1), \qquad u(-1) = u(1) = 0.$$

La soluzione è data dalla funzione $u(x) = 1 - |x|^q$.
Calcolare l'ordine di convergenza del metodo delle differenze finite in corrispondenza ai seguenti valori di $q$:

$$q = 2, 2.5, 3, 3.5, 4.$$

Cosa succede se $q < 2$? Per esempio per $q = 1$ o $q = 3/2$?

Usare le seguenti scelte per $N$:

```
N=[4 8 16 32 64 128 256 512];
N=[4 8 16 32 64 128 256 512]+1;
```

# Esercizio 4

Si consideri l'equazione differenziale

$$-u''(x) = 12x^2 \text{ per } x \in (-1, 1) \quad u(-1) = u(1) = 0.$$

La soluzione è $u(x) = 1 - x^4$.
Calcolare la soluzione con il metodo delle differenze finite in corrispondenza dei seguenti valori di `N`

```
N=[10 20 40 80 160 320 640 1280];
N=[N 2500 5000 10000 20000 40000 80000 1.e5 1.2e5
1.5e5];
```

Valutare per ciascun valore di `N` l'errore relativo e riportare gli errori in funzione di `N` in un grafico in scala bilogaritmica insieme all'andamento teorico.

# eye

Il comando `eye` serve per costruire la matrice identità, ossia

$$I = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

`eye(N)` fornisce la matrice identità di dimensione $\mathtt{N} \times \mathtt{N}$
`eye(M,N)` oppure `eye([M,N])` è una matrice di dimensione $\mathtt{M} \times \mathtt{N}$ con `1` sulla diagonale principale e `0` altrove.
`eye(SIZE(A))` è una matrice identità con le stesse dimensioni di `A`.

# zeros e ones

### `ones` e `zeros`

I comandi `ones` e `zeros` permettono di costruire un array che ha componenti tutte uguali ad 1 oppure a 0.
`b=ones(10,1)` costruisce il vettore colonna $b$ di 10 componenti tutte uguali a 1.
`c=zeros(3,3)` costruisce una matrice $3 \times 3$ di elementi tutti uguali a 0.

# diag

Il comando **diag** ha due funzioni diverse:

### Costruzione di una matrice con una sola diagonale non nulla

Sia `V` un vettore di `N` componenti.
`diag(V,K)` è una matrice quadrata di ordine `N+ABS(K)` che ha gli elementi di `V` sulla diagonale `K`-esima.

se `K`$= 0$ è la diagonale principale
se `K`$> 0$ si trova sopra la diagonale principale
se `K`$< 0$ si trova sotto la diagonale principale

### Estrazione di una diagonale da una matrice

Sia `A` una matrice di dimensione $n \times n$ il comando

```
d=diag(A,k)
```

fornisce il vettore
$d = [a_{1,1+k}, \dots, a_{n-k,n}]^t$ se $k \geq 0$
$d = [a_{1+k,1}, \dots, a_{n,n-k}]^t$ se $k < 0$.

# Formato `sparse`

Il formato sparse è utilizzato in Matlab per ridurre i costi di memorizzazione della matrice.

`S=sparse(A)` converte la matrice in formato full in una matrice in formato sparse tenendo in memoria solo gli elementi diversi da zero.

`S=sparse(m,n)` genera una matrice in formato sparse con tutti gli elementi nulli.

`S = sparse(i,j,s,m,n)` usa i vettori `i`, `j` e `s` per costruire la matrice di dimensione $m \times n$ tale che `S(i(k),j(k)) = s(k)`.

Il comando `spy(A)` mostra in un grafico quali sono gli elementi di $A$ non nulli ed il loro numero `nnz`.

# Il comando `spdiags`

Il comando `spdiags` generalizza il comando `diag`.

Sono disponibili quattro operazioni differenti.

- `B = spdiags(A)` estrae tutte le diagonali non nulle dalla matrice $A$. Le $p$ colonne di $B$ sono le diagonali di $A$. `[B,d] = spdiags(A)` fornisce anche il vettore d di lunghezza $p$, i cui valori specificano le diagonali di $A$.
- `B = spdiags(A,d)` estrae le diagonali specificate da $d$.
- `A = spdiags(B,d,A)` sostituisce le diagonali specificate da $d$ con le colonne di $B$.
- `A = spdiags(B,d,m,n)` crea una matrice sparsa $m \times n$ prendendo le colonne di $B$ e mettendole al posto delle diagonali specificate da $d$.