Metodi di Krylov

Metodi per la soluzione numerica di sistemi di equazioni lineari non simmetriche applicabili in un sottospazio di Krylov.

Fabio Archenti Fabio Camagni Andrea Favero
Lorenzo Fiamingo Stefano Gallivanone Nazariy Nashkolnyy

29/04/2020

Bibliografia

[1] Quarteroni, Saleri, and Gervasio.
Calcolo Scientifico: Esercizi e problemi risolti con MATLAB e Octave.
UNITEXT. Springer Milan, 2012.

[2] Trefethen, and Bau.
Numerical Linear Algebra.
SIAM, 1997.

[3] Telichevesky, Kundert, and White.
Efficient Steady-State Analysis based on Matrix-Free Krylov-Subspace Methods.
Scientific report, June 1995.

Sommario

Introduzione

- I metodi di Krylov risolvono sistemi lineari $Ax = b$, dove:
  - $A \in \mathbb{C}^{n \times n}$ è una matrice quadrata
  - $x \in \mathbb{C}^n$ è il vettore incognita
  - $b \in \mathbb{C}^n$ è il vettore dei termini noti
- Ad ogni passo dell'iterazione viene calcolata una soluzione approssimata x_k appartenente al sottospazio di Krylov:
- Si definisce **sottospazio di Krylov** il sottospazio generato:

$$\mathcal{K}_n(A, b) = span\{b, Ab, A^2b, \ldots, A^{n-1}b\},\ n \geq 1$$

.

Introduzione (II)

- x_0 è la **guess** iniziale al passo 0.
- Definiamo la **soluzione** del sistema cercata

$$x^* = A^{-1}b$$

- Definiamo il **residuo** del sistema al passo k

$$r_k = b - Ax_k$$

- Definiamo l'**errore** del sistema al passo k

$$e_k = x_k - x^*$$

Possiamo notare come $r_k = -A^{-1}e_k$.

Metodi di discesa

- Data A matrice simmetrica e definita positiva, possiamo definire la funzione (detta **forma quadratica**) $\phi : \mathbb{R}^n \to \mathbb{R}$

$$\phi(\mathrm{x}) = \frac{1}{2}\mathrm{x}^T A\mathrm{x} - \mathrm{x}^T \mathrm{b}$$

- Questa funzione (se le ipotesi su A sono verificate) è una funzione convessa e ammette un unico punto x^* di **minimo assoluto**.

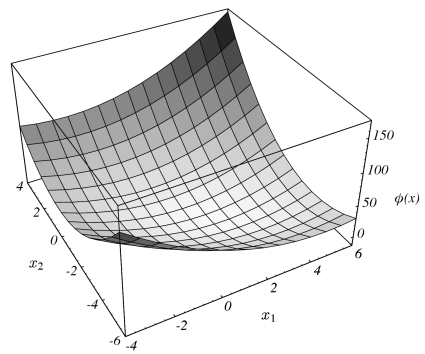

Figura 1: Forma quadratica per una matrice definita positiva

Metodi di discesa (II)

- Calcolo del **gradiente** di $\phi(\mathrm{x})$: $\nabla\phi(\mathrm{x}) = A\mathrm{x} - \mathrm{b}$ (con $\nabla(\mathrm{b}^T\mathrm{x}) = \mathrm{b}$ e $\nabla(\mathrm{x}^T A\mathrm{x}) = 2A\mathrm{x}$).
- Nel punto x^* il gradiente si annulla: $\nabla\phi(\mathrm{x}^*) = A\mathrm{x}^* - \mathrm{b} = 0$. Risolvere il problema di minimo su $\phi(\mathrm{x})$ (trovando x^*) equivale quindi a risolvere il sistema $A\mathrm{x} = \mathrm{b}$.
- Assegnato un $\mathrm{x}_0 \in \mathbb{R}^n$, si procede per ogni k fino a convergenza:
  1. determinando una direzione di discesa $\mathrm{d}_k \in \mathbb{R}^n$
  2. determinando un passo $\alpha_k \in \mathbb{R}$
  3. ponendo $\mathrm{x}_{k+1} = \mathrm{x}_k + \alpha_k \mathrm{d}_k$
- L'approccio sarà lo stesso per il metodo del Gradiente e per il metodo del Gradiente Coniugato, ma sceglieremo direzioni di discesa d_k diverse per ogni metodo.

Metodo del Gradiente

- Per il metodo del Gradiente la direzione d_k è il residuo $r_k = b - Ax_k$.
- $\nabla\phi(x) = Ax - b$: $d_k = r_k = -\nabla\phi(x)$.
- Si può dimostrare che il passo di discesa ottimale è $\alpha_k = \frac{d_k^T r_k}{d_k^T A d_k}$.
- Si aggiorna la soluzione: $x_{k+1} = x_k + \alpha_k d_k$.
- E infine si aggiorna il residuo: $r_{k+1} = r_k - \alpha_k A d_k$.

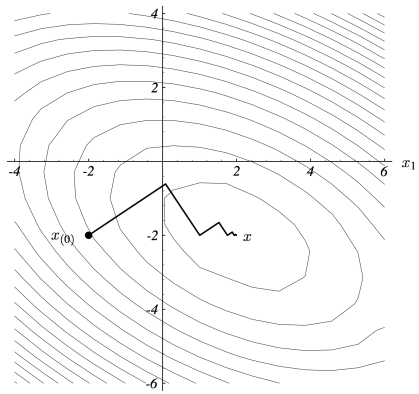

Figura 2: Iterazioni del metodo del gradiente

Metodo del Gradiente Coniugato - Passo di discesa

- Il **passo di discesa** più conveniente (che minimizza $\phi(x_{k+1})$) è

$$\alpha_k = \frac{d_k^T r_k}{d_k^T A d_k}$$

.

- Dimostrazione:
  - Sostituendo la soluzione al passo $k+1$ $x_{k+1} = x_k + \alpha_k d_k$ nella forma quadratica

$$\phi(x_{k+1}) = \phi(x_k + \alpha_k d_k) = \frac{1}{2}\left(d_k^T A d_k\right)\alpha_k^2 - d_k^T \underbrace{(b - Ax_k)}_{=r_k}\alpha_k + \frac{1}{2}x_k^T A x_k - x_k^T b$$

  - Derivando e imponendo la condizione di stazionarietà

$$\frac{d\phi}{d\alpha_k}(x_k + \alpha_k d_k) = \left(d_k^T A d_k\right)\alpha_k - d_k^T r_k = 0$$

  - Ricaviamo α_k ottenendo la formula cercata.

Calcolo del residuo

- Calcoliamo il **residuo** $r_k = b - Ax_k$:

$$r_{k+1} = r_k - \alpha_k A d_k$$

.

- Dimostrazione:
  - $r_{k+1} = -Ae_{k+1}$
  - $r_{k+1} = -A(e_k + \alpha_k d_k)$
  - $r_{k+1} = r_k - \alpha_k A d_k$
- Questa formula, in caso di matrici di grandi dimensioni, potrebbe essere soggetta ad errori di troncamento. Una buona soluzione potrebbe essere l'utilizzare $r_k = b - Ax_k$ dopo alcune iterazioni.

Direzione di discesa

- Utilizziamo l'**ortogonalizzazione di Gram-Schmidt** per ottenere direzioni $\{\mathrm{d}_k\}$ A-ortogonali a partire dai residui $\{\mathrm{r}_k\}$
  - $\mathrm{d}_0 = \mathrm{r}_0$
  - $\mathrm{d}_{k+1} = \mathrm{r}_{k+1} + \sum_{h=0}^{k} \beta_{kh} \mathrm{d}_h$
- Dato che $A\mathcal{K}_k$ è incluso in $\mathcal{K}_{k+1}$, il fatto che r_{k+1} sia ortogonale a $\mathcal{K}_{k+1}$ implica che r_{k+1} sia A-ortogonale a $A\mathcal{K}_k$, e quindi a tutte le precedenti direzioni di discesa d, esclusa d_k.
- Di conseguenza **non sarà necessario memorizzare le vecchie direzioni** per assicurare la A-ortogonalità delle nuove.
- La **direzione di discesa** diviene

$$\mathrm{d}_{k+1} = \mathrm{r}_{k+1} + \beta_{k+1} \mathrm{d}_k$$

dove

$$\beta_{k+1} = \frac{\mathrm{r}_{k+1}^T \mathrm{r}_{k+1}}{\mathrm{r}_k^T \mathrm{r}_k}$$

Passo di discesa (II)

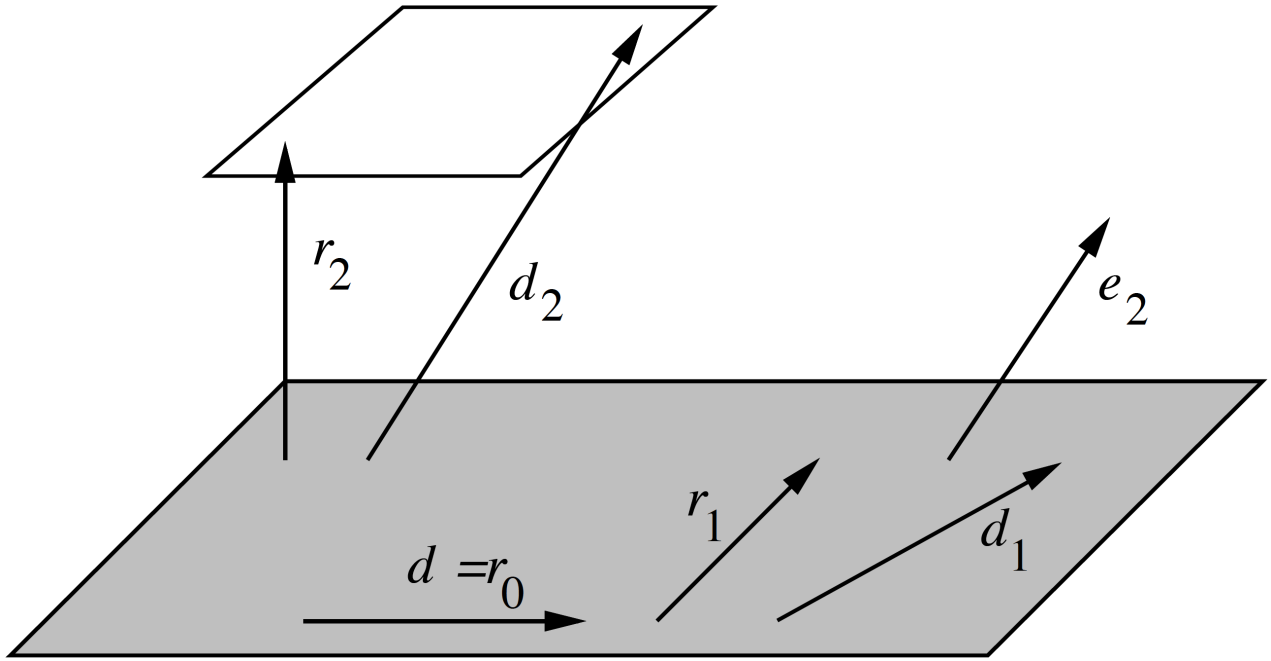

Figura 3: Il piano scuro è il sottospazio $\mathcal{K}_2$. Notiamo che r_2 e d_2 puntano su un piano parallelo a $\mathcal{K}_2$ (come conseguenza dell'ortogonalizzazione)

- Dalla Figura 3 $d_k^T r_k = r_k^T r_k$, riscriviamo il passo di discesa come

$$\alpha_k = \frac{r_k^T r_k}{d_k^T A d_k}$$

Velocità di convergenza

- Il metodo del gradiente coniugato converge al massimo in n iterazioni (in aritmetica esatta), e si ottiene che

$$\|\mathrm{e}_{(k)}\|_A \leq \frac{2c^k}{1+c^{2k}}\|\mathrm{e}_{(0)}\|_A$$

$$\|\mathrm{e}_{(k)}\|_A \leq 2c^k\|\mathrm{e}_{(0)}\|_A$$

$$c = \frac{\sqrt{K(A)}-1}{\sqrt{K(A)}+1}.$$

$K(A)$ è il numero di condizionamento in norma 2 di A

- Come metodo iterativo, viene arrestato quando il residuo relativo $E_{(k)} = \frac{\|\mathrm{r}_{(k)}\|}{\|\mathrm{r}_{(0)}\|}$ è minore di una tolleranza data.

Precondizionamento (I)

- La velocità di convergenza dipende quindi dal numero di condizionamento $K(A)$.
- Per accelerare la convergenza dobbiamo diminuire il numero di condizionamento ($K(P^{-1}A) \ll K(A)$)
- Possiamo risolvere il **sistema precondizionato** (equivalente a quello dato, $A\mathrm{x} = \mathrm{b}$)

$$M^{-1}A\mathrm{x} = M^{-1}\mathrm{b}$$

- In generale se M e A sono simmetriche e definite non è detto che lo sia $M^{-1}A$, ma per ogni M simmetrica, definita positiva esiste almeno una matrice P tale che $PP^T = M$.
- Dato che $M^{-1}A$ e $P^{-1}AP^T$ hanno gli stessi autovalori possiamo scrivere il sistema equivalente

$$P^{-1}AP^{-T}\hat{\mathrm{x}} = P^{-1}\mathrm{b},\ \hat{\mathrm{x}} = P^T\mathrm{x}$$

- $P^{-1}AP^{-T}$ è simmetrica e definita positiva.

Precondizionamento (II)

- Per tornare al caso del primo sistema precondizionato sfruttiamo l'equazione $M^{-1} = P^{-T}P^{-1}$
- Scegliendo M stiammo operando un *trade-off* tra la diminuzione del numero di condizionamento e il costo computazionale della risoluzione del sistema $\mathrm{s}_{(\mathrm{k})} = M^{-1}\mathrm{r}_{(\mathrm{k})}$ per trovare il residuo precondizionato $\mathrm{s}_{(\mathrm{k})}$.
- Esempio di algoritmi applicabili per calcolare M sono:
  - Precondizionamento di Jacobi
  - Precondizionamento incompleto di Cholesky
- La convergenza del metodo precondizionato è più veloce, poichè $K(A)$ è stato sostituito con $K(P^{-1}A)$:

$$\|\mathrm{e}_{(k)}\|_A \leq \frac{2c^k}{1+c^{2k}}\|\mathrm{e}_{(0)}\|_A, c = \frac{\sqrt{K(P^{-1}A)}-1}{\sqrt{K(P^{-1}A)}+1}.$$

Metodo del gradiente coniugato (senza precondizionamento)

- $\mathsf{d}_0 = \mathsf{r}_0 = \mathsf{b} - A\mathsf{x}_0$
- $\alpha_k = \frac{\mathsf{r}_k^T \mathsf{r}_k}{\mathsf{d}_k^T A \mathsf{d}_k}$
- $\mathsf{x}_{k+1} = \mathsf{x}_k + \alpha_k \mathsf{d}_k$
- $\mathsf{r}_{k+1} = \mathsf{r}_k - \alpha_k A \mathsf{d}_k$
- $\beta_{k+1} = \frac{\mathsf{r}_{k+1}^T \mathsf{r}_{k+1}}{\mathsf{r}_k^T \mathsf{r}_k}$
- $\mathsf{d}_{k+1} = \mathsf{r}_{k+1} + \beta_{k+1} \mathsf{d}_k$

Implementazione Matlab (senza precondizionamento)

Dati come input una matrice A, il vettore dei termini noti b, una guess iniziale x, un numero massimo di iterazioni kmax e una tolleranza e

```
k = 0;
r = b-A*x; d = r;
delta = r'*r; delta0 = delta;
while k < kmax && delta > e^2*delta0
    q = A*d;
    alpha = delta/(d'*q);
    x = x+alpha*d;
    r = r-alpha*q;
    deltaold = delta;
    delta = r'*r;
    beta = delta/deltaold;
    d = r+beta*d;
    k=k+1;
end
```

Metodo del gradiente coniugato (con precondizionamento)

- $\mathsf{r}_0 = \mathsf{b} - A\mathsf{x}_0$
- $\mathsf{d}_0 = M^{-1}\mathsf{r}_0$
- $\alpha_k = \frac{\mathsf{r}_k^T M^{-1}\mathsf{r}_k}{\mathsf{d}_k^T A\mathsf{d}_k}$
- $\mathsf{x}_{k+1} = \mathsf{x}_k + \alpha_k \mathsf{d}_k$
- $\mathsf{r}_{k+1} = \mathsf{r}_k - \alpha_k A\mathsf{d}_k$
- $\beta_{k+1} = \frac{\mathsf{r}_{k+1}^T M^{-1}\mathsf{r}_{k+1}}{\mathsf{r}_k^T M^{-1}\mathsf{r}_k}$
- $\mathsf{d}_{k+1} = M^{-1}\mathsf{r}_{k+1} + \beta_{k+1}\mathsf{d}_k$

Implementazione Matlab (con precondizionamento)

Dati come input una matrice A, il vettore dei termini noti b, una guess iniziale x, la matrice di precondizionamento M, un numero massimo di iterazioni kmax e una tolleranza e

```
k = 0
r = b-A*x;
d = M\r; %%piu' performante di inv(M)*r
delta = r'*r; delta0 = delta;
while k < kmax && delta > e^2*delta0
    q = A*d;
    alpha = delta/(d'*q);
    x = x+alpha*d;
    r = r-alpha*q;
    s = M\r;
    deltaold = delta;
    delta = r'*s;
    beta = delta/deltaold;
    d = s+beta*d;
    k=k+1;
end
```

Considerazioni finali

Vantaggi

- Possibilità di implementare questi metodi in versione *matrix-free*, ovvero ad ogni iterazione basta solo calcolare un prodotto matrice-vettore, e altre operazioni meno computazionalmente costose.
- Convergenza del metodo del gradiente coniugato in meno iterazioni rispetto alla dimensione n della matrice A (calcolando invece la soluzione attraverso Cramer o con i metodi diretti ci sono operazioni matrice-matrice che possono risultare molto dispendiose).
- Se la matrice è **sparsa**, è possibile memorizzare i suoi elementi a un costo limitato ($O(n)$ invece di $O(n^2)$). Non è detto però i suoi fattori siano matrici sparse, quindi non è consigliabile applicare metodi di fattorizzazione.

Svantaggi

- In quanto metodi iterativi, sono soggetti a errori di troncamento.

Confronto Gradiente - Gradiente Coniugato

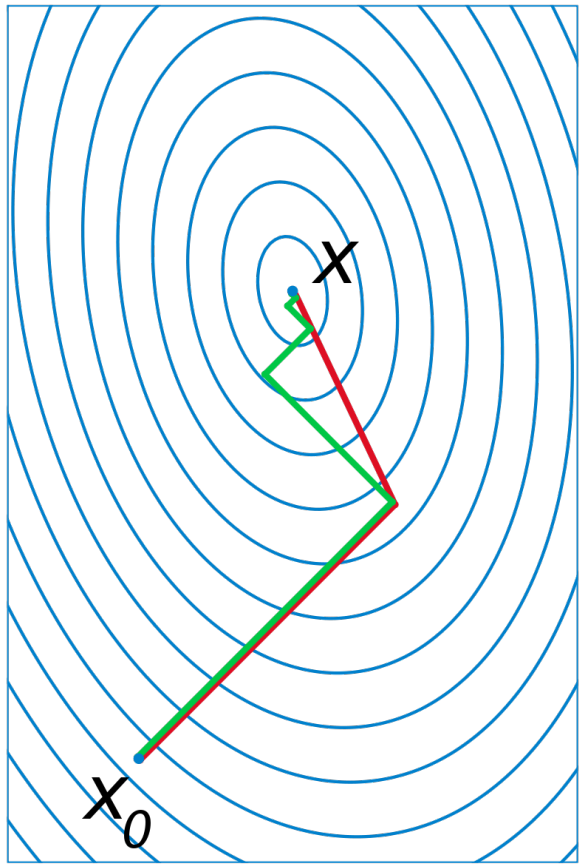

Figura 4: Confronto tra il numero di iterazioni dei due metodi

Come si può osservare dalla figura, il metodo del Gradiente (in verde) esegue molte più iterazioni rispetto alle (sole!) due del metodo del Gradiente Coniugato.

Metodo GMRES

- **GMRES** (General Minimal RESidual) è uno dei metodi di Krylov utilizzato per risolvere il sistema lineare $Ax = b$ tramite il sottospazio $\mathcal{K}_n$ in modo tale da rendere minima la norma Euclidea del residuo:

$$\|b - Ax_n\|_2.$$

- Sia K_n la matrice *mxn* di Krylov, composta da spazi colonne $A\mathcal{K}_n$. Inanzitutto si moltiplica K_n per A, per cui il problema diventa trovare un vettore $c \in \mathbb{C}^n$, tale per cui il residuo $\|AK_nc - b\|_2$ sia minimo.
- Questa formula è risolvibile con fattorizzazione QR, ma AK_n diventa **numericamente instabile** al crescere di n, quindi si utilizza **l'iterazione di Arnoldi** per costruire una base di vettori ortonormali.

L'iterazione di Arnoldi (I)

- Si definisce Q_n come la matrice, i cui vettori formano una base di $\mathcal{K}_\mathrm{n}$. Il problema consiste ora nella ricerca di y $\in \mathbb{C}^n$ tale per cui:

$$\|AQ_n\mathrm{y} - \mathrm{b}\|_2 = \textit{minimo},$$

  riducendo così le dimensioni del problema da mxn a $(n+1)xn$.
- L'iterazione di Arnoldi dimostra che $AQ_n = Q_{n+1}\mathcal{H}_n$ (con $\mathcal{H}_n$ la matrice alta-sinistra, $(n+1)xn$, della matrice di Hessenberg). Il problema quindi diventa:

$$\|Q_n\mathcal{H}_n\mathrm{y} - \mathrm{b}\|_2 = \textit{minimo}.$$

L'iterazione di Arnoldi (II)

- Moltiplicando entrambi i membri per Q_{n+1}^* si ottiene:

$$\|\mathcal{H}_n \mathsf{y} - Q_{n+1}^* \mathsf{b}\|_2 = \textit{minimo}.$$

- Si osserva che $Q_{n+1}^* \mathsf{b} = \|\mathsf{b}\| \mathsf{e}_1$, con $\mathsf{e}_1 = (1, 0, 0, \dots)^*$, quindi il problema finale è:

$$\|\mathcal{H}_n \mathsf{y} - \|\mathsf{b}\| \mathsf{e}_1\|_2 = \textit{minimo}.$$

Questa equazione è risolvibile con l'approssimazione ai minimi quadrati.

GMRES Polinomi

- Il metodo GMRES è utilizzabile anche nel calcolo polinomiale.
- Dato $p_n \in P_n$, con $p_n(0) = 1$ si definisce $x_n = q_n(A)\mathrm{b}$ e $r_n = p_n(A)\mathrm{b}$. Il problema consiste nel calcolare $p_n \in P_n$ tale che

$$\|r_n\| = \mathit{minimo}.$$

Velocità di convergenza

Quanto deve valere **n** (numero di iterazioni) per soddisfare la tolleranza?

$$\frac{\|r_n\|}{\|\mathrm{b}\|} < tolleranza$$

- Osservazione 1: convergenza monotona $\|r_{n+1}\| \leq \|r_n\|$.
- Osservazione 2: per $n \to \infty$, $\|r_\infty\| = 0$, a meno di errori di arrotondamento.
- Essendo $\|r_n\| = \|p_n(A)\|\|\mathrm{b}\|$ risulta che solo $p_n(A)$ influenza la velocità di convergenza:

$$\frac{\|r_n\|}{\|\mathrm{b}\|} \leq inf\,\|p_n(A)\|,$$

con $p_n \in P_n$.

Come inserire una formula

Una formula può essere inserita all'interno del testo così : $-\nabla\left(\varepsilon\nabla u\right)=f$ oppure centrata e numerata così:

$$-\nabla\cdot\left(\varepsilon\nabla u\right)=f \tag{1}$$

oppure centrata senza numerazione così :

$$-\nabla\cdot\left(\varepsilon\nabla u\right)=f$$

Posso fare riferimento alle formule numerate così : (1)

Come inserire una lista per punti

Ecco come inserire una lista per puti

- un punto
- un altro

oppure un elenco numerato

1. primo punto
2. secondo punto

Come inserire un pezzo di codice

Si può evidenziare **parte del testo** in questo modo. Si può inserire un comando matlab all'interno del testo in questo modo :
for i = 1 : 10, **disp** (i), **end**
si può inserire un un file contenete un codice matlab in questo modo :

```
for i = 1 : 10
  disp (i);
end
```

Si possono inserire solo alcune righe del file in questo modo :

```
for i = 1 : 10
  disp (i);
```

Come inserire una figura

Questo è un esempio di come inserire una figura

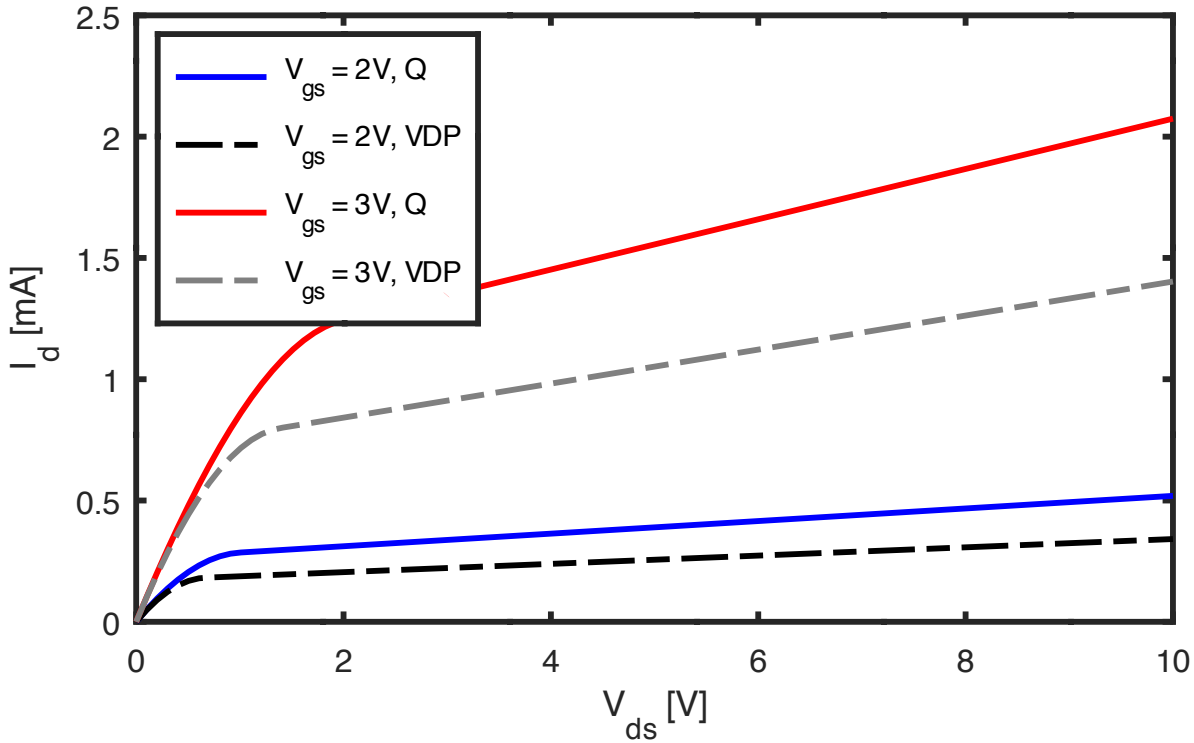

Figura 5: Questa è la didascalia

Se la figura ha un'etichetta la si può usare per fare riferimento alla figura nel testo : Figura 5

Altra documentazione

Questo è un esempio di come inserire un link ad un URL

- Altre informazioni utili
  - Il sito del LaTeX–project
    - Una lista di software offline ed online per creare documenti LaTeX
  - Un Wikibook su LaTeX
    - La sezione sulle presentazioni
    - La sezione sulle formule
  - Un forum con domande e risposte
  - La documentazione di overleaf
    - Un tutorial per iniziare in 30 minuti
  - Uno strumento per cercare simboli