# Lezione n° 4

Sommario:

- Ortogonalizzazione
- Determinazione di autovettori e autovalori
- Esercizi in preparazione all'esame

# ORTOGONALIZZAZIONI

In matematica, e in particolare in algebra lineare, l'ortogonalizzazione di Gram-Schmidt è un algoritmo che permette di ottenere un insieme di vettori ortogonali a partire da un generico insieme di vettori linearmente indipendenti in uno spazio vettoriale dotato di un prodotto scalare definito positivo.

Si definisce **proiezione ortogonale** $\mathbf{proj_u v}$ la funzione che proietta il vettore **v** in modo ortogonale su **u**:

$$\mathrm{proj}_u v = \frac{\langle v, u\rangle}{\langle u, u\rangle} u$$

Dove $\langle v, u\rangle$ viene definito il prodotto scalare tra v e u.

Il procedimento di Gram–Schmidt permette di costruire una base ortogonale $\mathbf{u_1}, \ldots, \mathbf{u_n}$ a partire da una base generica $\mathbf{v_1}, \ldots, \mathbf{v_n}$. Normalizzando quindi la base ortogonale si ottiene una base ortonormale dello spazio.

I primi due passi dell'algoritmo sono:

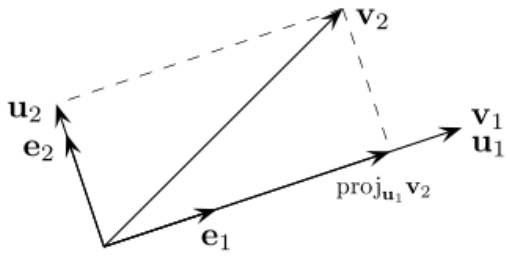
$$u_1 = v_1, \qquad e_1 = \frac{u_1}{\|u_1\|}$$

$$u_2 = v_2 - \mathrm{proj}_{u_1} v_2, \qquad e_2 = \frac{u_2}{\|u_2\|}$$

$$u_3 = v_3 - \mathrm{proj}_{u_1} v_3 - \mathrm{proj}_{u_2} v_3, \quad e_3 = \frac{u_3}{\|u_3\|}$$

$$\ldots$$

$$u_k = v_k - \sum_{j=1}^{k-1} \mathrm{proj}_{u_j} v_k \qquad e_k = \frac{u_k}{\|u_k\|}$$



```
%Esempio
```

Si propone ora di ortonormalizzare la seguente base di $\mathbb{R}^3$ attraverso il procedimento di Gram-Schmidt:

```
w1=[4 2 1];    w2=[1 5 2];    w3=[3 10 6];
```

Lascio inalterato il primo vettore

```
>> u1 = w1;
```

```
>> u2 = w2-dot(w2,u1)/norm(u1)^2*u1;
```

```
>> u3 = w3-dot(w3,u2)/norm(u2)^2*u2-dot(w3,u1)/norm(u1)^2*u1;
```

Si possono ottenere i relativi versori:

```
>> V1 = u1/norm(u1);
```

```
>> V2 = u2/norm(u2);
```

```
>> V3 = u3/norm(u3);
```

```
------------------------------clear--------------------------------
------------------------------clc----------------------------------
```

# Determinazione di autovettori e autovalori

Si definisce un **autovettore** di una trasformazione lineare tra spazi vettoriali è un vettore la cui immagine è il vettore stesso moltiplicato per uno scalare, detto **autovalore**.

Sia data la matrice

```
>> A = [1 6; 5 2];
```

NOTA BENE:

```
>> help eig
```

```
…
[V,D] = EIG(X) produces a diagonal matrix D of eigenvalues and a
full matrix V whose columns are the corresponding eigenvectors so
that X*V = V*D.
…
```

Il comando **eig** permette di calcolare gli autovalori di una matrice quadrata

```
>> X = eig(A);
```

Inoltre, il comando `eig` dà la possibilità di discriminare il risultato della funzione nel seguente modo

```
>> [Xa,Da] = eig(A);
```

dove `Da` è una matrice diagonale i cui elementi sono gli autovalori, mentre `Xa` è una matrice in cui le colonne contengono i corrispondenti autovettori.

Si può verificare che `inv(Xa)*A*Xa` mi permette di calcolare `Da`:

```
>> inv(Xa)*A*Xa;
```

Vediamo ora un esempio con una matrice simmetrica:

```
>> B = [5 4 2; 4 5 2; 2 2 2];
```

```
>> [Xb,Db] = eig(B);
```

Si osserva che nel caso della matrice simmetrica, oltre a valere ancora la relazione precedente, `inv(Xb)*B*Xb=Db`, vale anche `Xb'*B*Xb=Db` poiché in questo caso la trasposta è uguale all'inversa, `Xb'==inv(Xb)`.

**----------clear----------**
**----------clc----------**

# Esercizi di preparazione all'esame

```
%Esercizio 1 - Sistema Lineare
```

Dato il seguente sistema lineare parametrico in t

$$\begin{cases} 3x + (t+13)y = -4t - 16 \\ 2x + (t+2)z = 2 \\ (4-t)z - x = 9 - t \end{cases}$$

si vuole calcolare:

1. Il determinante di A;
2. la soluzioni del sistema al variare di t (in forma simbolica)
3. il valore della soluzione in funzione di t=2;

```
%Soluzione

>> syms t

>> A = [
3 t+13 0;
2 0 t+2;
-1 0 4-t;
];

>> B = [
-4*t-16;
2;
9-t;
];
```

Calcolo il determinante di `A`:

```
>> d = det(A);
```

Ottengo per il determinante un'equazione di secondo grado in `t`. Devo calcolare per quali valori di `t` questo si annulla (quindi il sistema non è di Cramer):

```
>> radici = solve(d,t);
```

Ora posso trovare la soluzione del sistema in funzione di tutti gli altri infiniti valori del parametro `t`:

```
>> Xt = A\B;
```

Per calcolare la soluzione del sistema per un particolare valore del parametro (per esempio `t=2`), è sufficiente applicare la funzione `subs` sulla soluzione `Xt` per il valore desiderato.

N.B.: il valore da passare alla subs non deve annullare il determinate di A (cioè non posso passare né 10 né -13)

```
>> valore = 2;

>> X = subs(Xt,t,valore);
```

Rimane da calcolare la soluzione del sistema per quei valori di `t` che annullano il determinante di `A` rendendo il sistema indeterminato, cioè con infinite soluzioni.

Usiamo Gauss-Jordan. Per prima cosa scrivo la matrice completa `C`:

```
>> C = [A B];
```

Considero `t=-13`, quindi vado a sostituire il valore -13 nella matrice C con il comando `subs`:

```
>> C1 = subs(C,t,-13);

>> R1 = rref(C1)

R1 =

     1     0     0    12

     0     0     1     2

     0     0     0     0
```

In questo caso la soluzione sarà caratterizzata dalla variabile libera `y`:

```
>> syms y

>> X1 = [R1(1,4); y; R1(2,4)]

X1 =

     12

      y

      2
```

Ora considero `t=10`:

```
>> C2 = subs(C,t,10);

>> R2 = rref(C2)

R2 =

    1.0000         0    6.0000    1.0000

         0    1.0000   -0.7826   -2.5652

         0         0         0         0
```

Analogamente a prima la soluzione sarà data dalla variabile libera `z`:

```
>> syms z

>> X2 = [-z*R2(1,3)+R2(1,4); -z*R2(2,3)+R2(2,4); z]

X2 =

           1 - 6*z

 (18*z)/23 - 59/23

                z


      ------------------------------clear---------------------------------
      ------------------------------clc---------------------------------
```

```
%Esercizio 2 - Geometria dello spazio
```

Costruire una funzione che prenda in ingresso le coordinate di un punto P espresse come vettore riga e i coefficienti di due equazioni che descrivono una retta (a1,b1,c1,d1), (a2,b2,c2,d2) e che restituisca in uscita l'espressione simbolica del piano che contiene entrambi.

```
%Soluzione

function [piano] = esame(P,a1,b1,c1,d1,a2,b2,c2,d2)

Dichiaro le variabili simboliche

syms x y z k

%Scrivo le due equazioni della retta (meglio, dei due piani che si
%intersecano) sfruttando le variabili simboliche appena dichiarate e i
%coefficienti in ingresso alla funzione.

eq1 = a1*x+b1*y+c1*z+d1;

eq2 = a2*x+b2*y+c2*z+d2;

%Scrivo l'equazione del fascio di rette sfruttando le due equazioni
%precedenti e la variabile simbolica k di cui devo trovare il valore.

eq_fascio = eq1+k*eq2;

%Sostituisco nell'equazione del fascio di rette le coordinate del punto P
%ottenendo così un'equazione nella sola variabile simbolica k.

eq_k = subs(eq_fascio,{x,y,z},{P(1,1),P(1,2),P(1,3)});

%Utilizzo il comando solve per risolvere l'equazione trovata al passo
%precedente e trovare quindi il valore di k

valore_k = solve(eq_k,k);

%Con il comando subs vado a sostituire il valore di k nell'equazione del
%piano e ottengo l'equazione che cercavo

Piano = subs(eq_fascio,k,valore_k);

return

end
```

testo la funzione in un esempio

P = [2 -1 1]

(2x + 3y+1=0 2y- z-2 =0)

Soluzione 10•x + 19•y - 2•z + 1 = 0