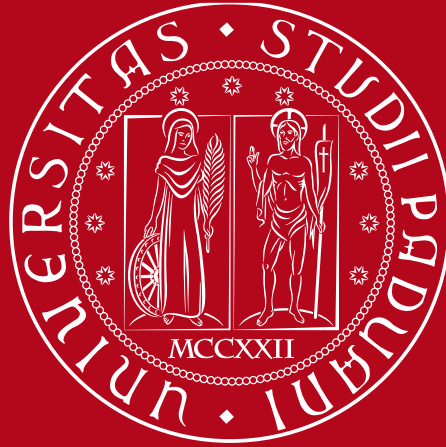


1222 \* 2022  
**800**  
ANNI



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

# Strumenti di analisi e progettazione di meccanismi tramite Python

## Laureandi

Alberto Feltre 1195425

Filippo Scudella 2033483

Steven Scremin 2034939

## Relatore

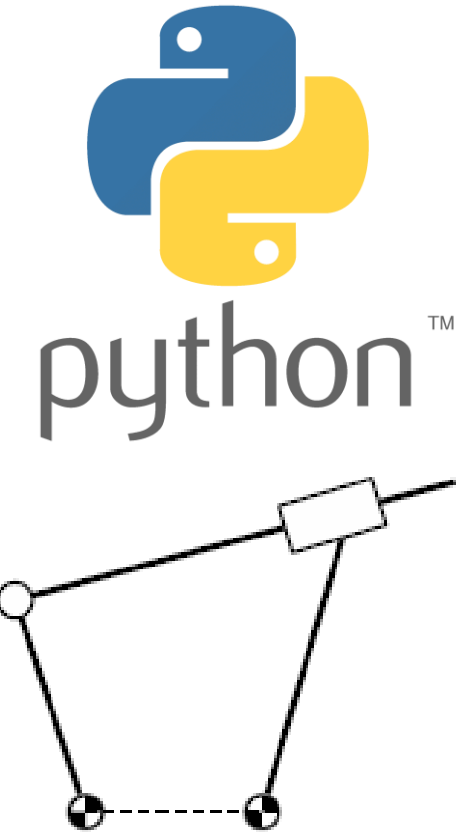
Prof. Paolo Boscariol

# Scopo

sviluppo e utilizzo di strumenti di analisi e progettazione per meccanismi articolati piani con Python

## Meccanismo articolato piano

Meccanismo in cui i corpi hanno moto piano



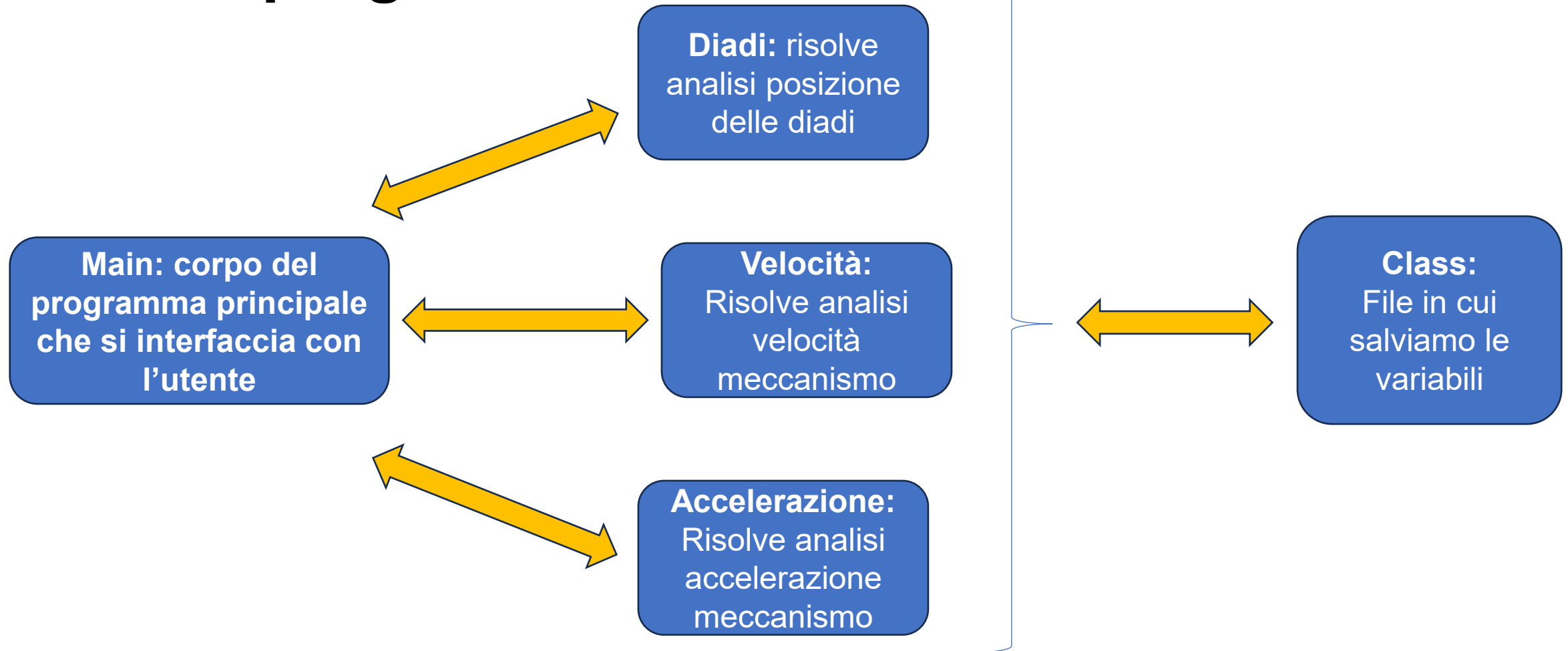
# Python:

Python è un linguaggio di programmazione moderno, dalla sintassi semplice e potente, con svariati ambiti di applicazione. Supporta sia la programmazione procedurale, sia la programmazione ad oggetti.

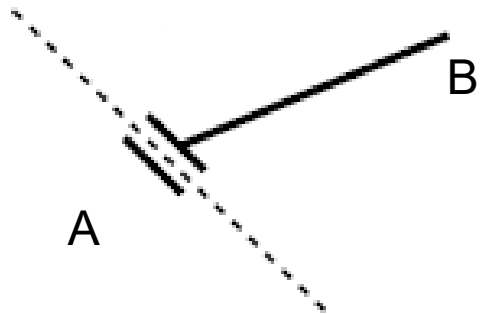
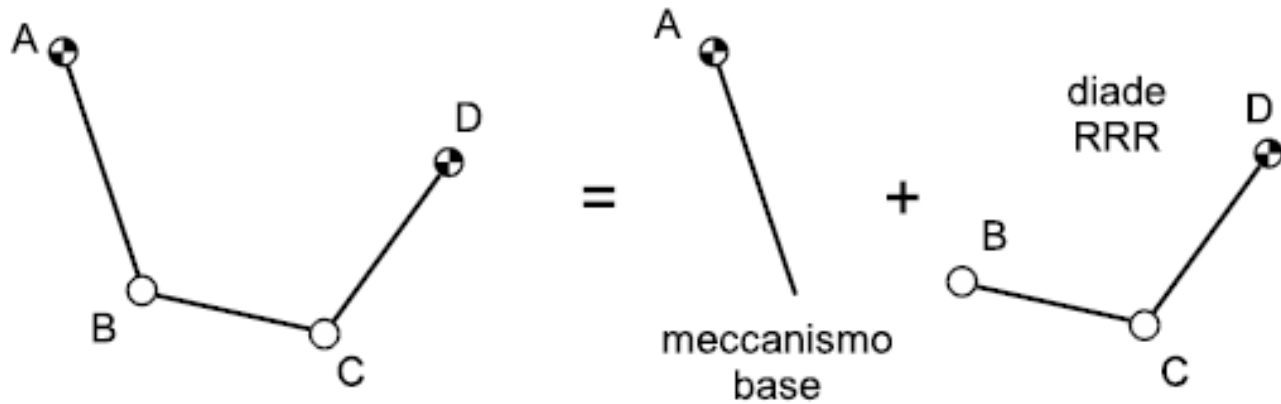
## Librerie usate di Python:

- **Numpy:** permette di usare e lavorare con array, matrici, ecc...
- **Matplotlib:** libreria utilizzata per disegnare il meccanismo finale
- **Math:** definisce alcune funzioni di base e costanti matematiche

# Struttura programma

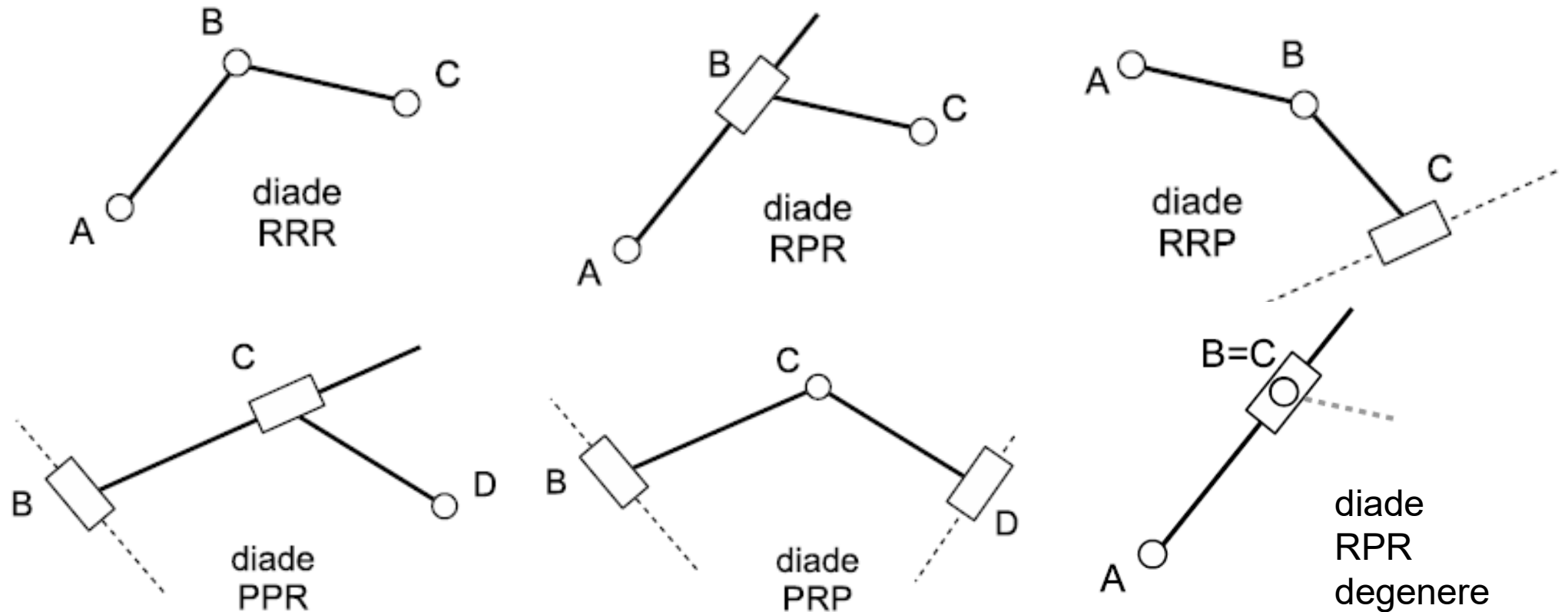


# Analisi di posizione: meccanismo base



```
def meccanismo_base(Ax, Ay, z1, f1): #MECCANISMO BASE
    A = np.array([Ax, Ay])
    B = np.array([0, 0])
    B = A + (z1 * sin_cos(f1))
    return B[0], B[1]
```

# Analisi di posizione: diadi



# Analisi di posizione

## diade PRP

```
def diade_PRP(Ax, Ay, Ex, Ey, a, b, g, d, z2, z3):
```

```
    A = np.array([Ax, Ay])
```

```
    E = np.array([Ex, Ey])
```

```
    zx = modulo(A, E)
```

```
    fx = fase(A, E)
```

```
    if(Ex >= 0):
```

```
        f1 = m.pi - a
```

```
        f2 = f1 - m.pi + b
```

```
        f4 = m.pi + d
```

```
        f3 = f4 + g
```

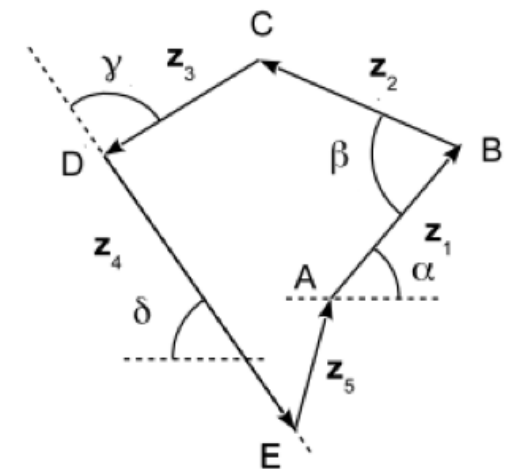
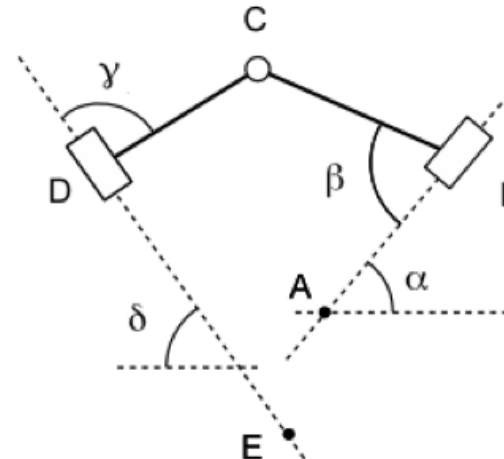
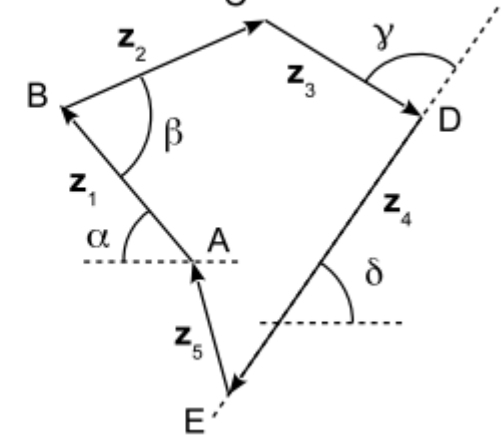
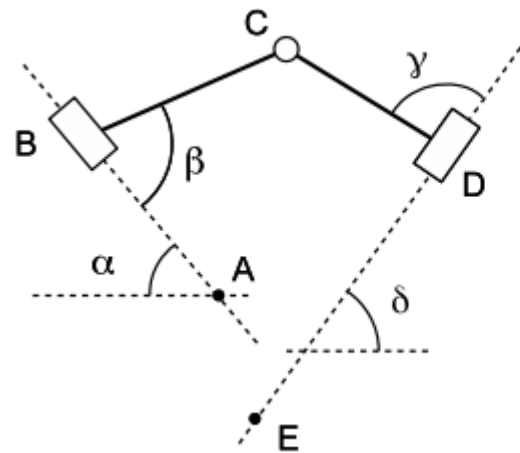
```
    elif(Ex < 0):
```

```
        f1 = a
```

```
        f2 = f1 + m.pi - b
```

```
        f4 = (2 * m.pi) - d
```

```
        f3 = f4 - g
```



# Analisi di posizione

$$\begin{cases} z_1 \cos \varphi_1 + z_2 \cos \varphi_2 + z_3 \cos \varphi_3 + z_4 \cos \varphi_4 = 0 \\ z_1 \sin \varphi_1 + z_2 \sin \varphi_2 + z_3 \sin \varphi_3 + z_4 \sin \varphi_4 = 0 \end{cases}$$

Equazione di chiusura

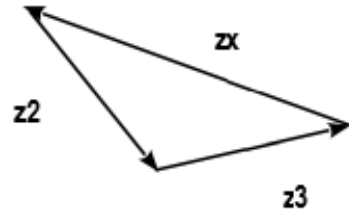
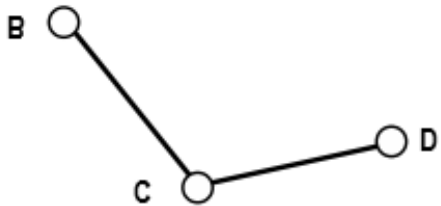
## diade RPP

```
incognite = np.array([[m.cos(f3), m.cos(f4)], [m.sin(f3), m.sin(f4)]])
noti = np.array([[-z2 * m.cos(f2) - zx * m.cos(fx)], [-z2 * m.sin(f2) - zx * m.sin(fx)]])
result = np.linalg.solve(incognite, noti)
z3 = result[0][0]
z4 = result[1][0]
C = B + (z2 * sin_cos(f2))
D = C + (z3 * sin_cos(f3))
f2 = angoli(f2)
f3 = angoli(f3)
f4 = angoli(f4)
return C[0], C[1], D[0], D[1], f2, f3, f4, z3, z4
```

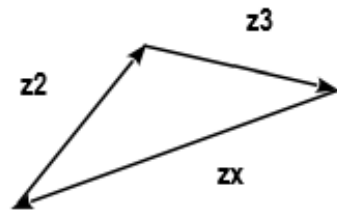
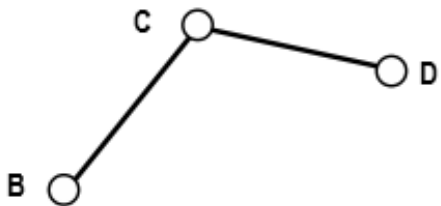


# Problema della configurazione

Configurazione gomito basso o a sinistra '1'

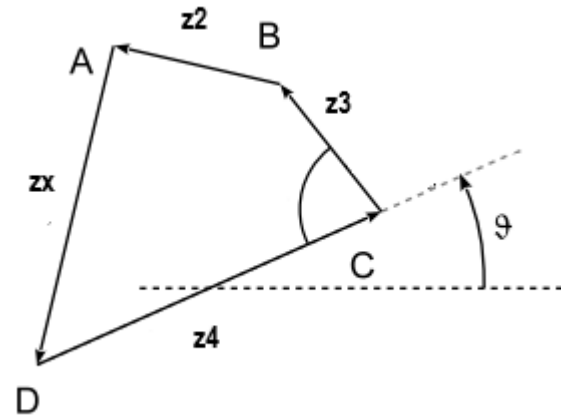
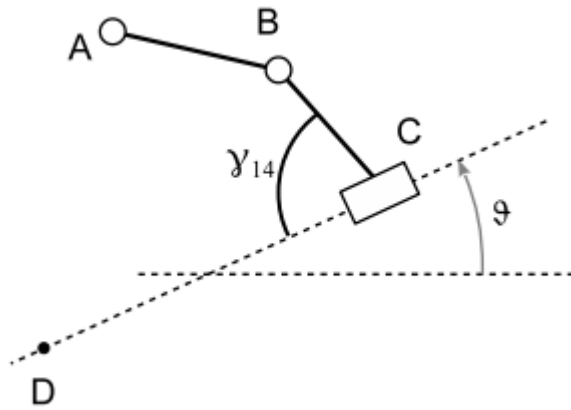


```
if(configurazione == "1"):
    f2 = fx - m.pi - g2x
elif(configurazione == "2"):
    f2 = fx - m.pi + g2x
```



Configurazione gomito alto o a destra '2'

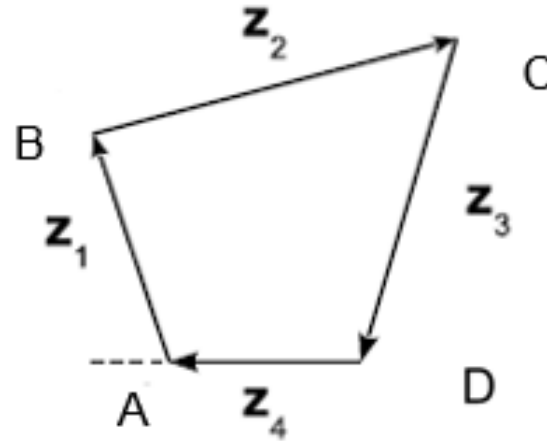
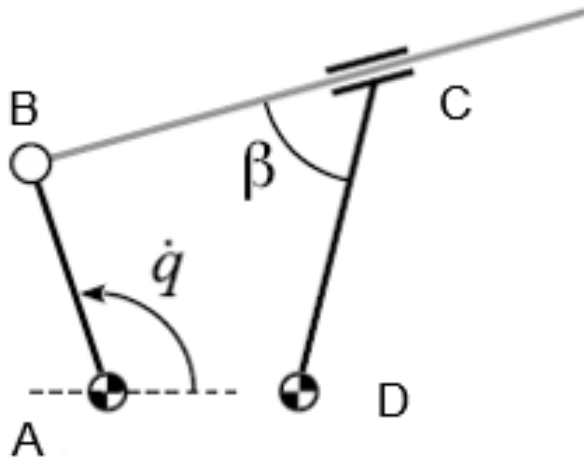
# Problema della configurazione



$$\psi_2 = m.\pi - m.\text{asin}((-z_3 * m.\sin(\psi_3) - z_x * m.\sin(\psi_{ix}))/z_2)$$

$$\psi_2 = m.\text{asin}((-z_3 * m.\sin(\psi_3) - z_x * m.\sin(\psi_{ix}))/z_2)$$

# Analisi di Velocità



Segmento AB è il meccanismo base.  
 $|z_1| = AB$  modulo vettore  $z_1$   
 $q = \varphi_1$  fase del vettore  $z_1$   
 $\dot{q} =$  velocità angolare vettore  $z_1$

$$\begin{cases} \sum_i (\dot{z}_i \cos \varphi_i - z_i \sin \varphi_i \dot{\varphi}_i) = 0 \\ \sum_i (\dot{z}_i \sin \varphi_i + z_i \cos \varphi_i \dot{\varphi}_i) = 0 \end{cases}$$

Derivata dell'equazione di chiusura

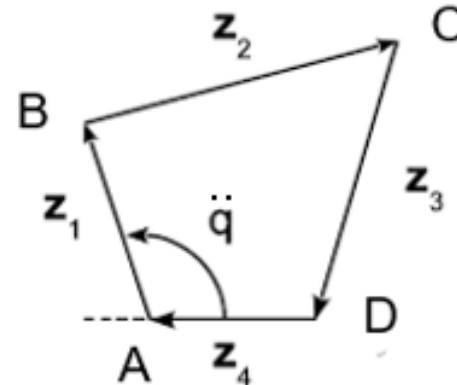
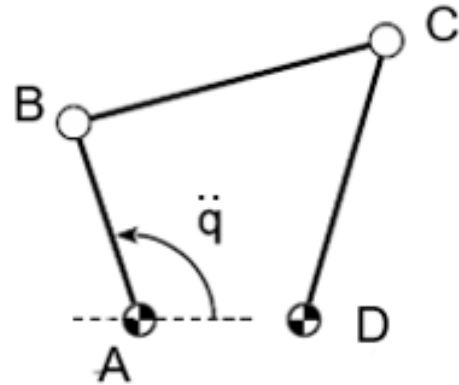
# Analisi di Velocità

## Matrice associata di Velocità

$$\begin{bmatrix} \cos \varphi_2 & -z_2 \sin \varphi_2 - z_3 \sin \varphi_3 \\ \sin \varphi_2 & z_2 \cos \varphi_2 + z_3 \cos \varphi_3 \end{bmatrix} \begin{bmatrix} \dot{z}_2 \\ \dot{\varphi}_3 \end{bmatrix} = \begin{bmatrix} -z_1 \sin \varphi_1 \\ z_1 \cos \varphi_1 \end{bmatrix} \dot{\varphi}_1$$

```
if(tipo_diade == "2"):  
    noti = np.array([[Diade.z1 * m.sin(Diade.f1) * Diade.f_1], [-Diade.z1 * m.cos(Diade.f1) * Diade.f_1]])  
    incognite = np.array([[m.cos(Diade.f2), -(Diade.z2 * m.sin(Diade.f2) + Diade.z3 * m.sin(Diade.f3))],  
                          [m.sin(Diade.f2), (Diade.z2 * m.cos(Diade.f2) + Diade.z3 * m.cos(Diade.f3))]])  
    risultato = np.linalg.solve(incognite, noti)  
    Diade.z_2 = risultato[0][0]  
    Diade.f_2 = risultato[1][0]  
    Diade.f_3 = Diade.f_2
```

# Analisi Accelerazione



$$\begin{cases} \sum_i (\ddot{z}_i \cos \varphi_i - z_i \sin \varphi_i \ddot{\varphi}_i - 2\dot{z}_i \sin \varphi_i \dot{\varphi}_i - z_i \cos \varphi_i \dot{\varphi}_i^2) = 0 \\ \sum_i (\ddot{z}_i \sin \varphi_i + z_i \cos \varphi_i \ddot{\varphi}_i + 2\dot{z}_i \cos \varphi_i \dot{\varphi}_i - z_i \sin \varphi_i \dot{\varphi}_i^2) = 0 \end{cases}$$

Equazione di chiusura  
relativa all'accelerazione

# Analisi Accelerazione

$$\begin{bmatrix} \ddot{z}_i \cos \varphi_i \\ \ddot{z}_i \sin \varphi_i \end{bmatrix} = \text{accelerazione relativa}$$

$$\begin{bmatrix} -z_i \sin \varphi_i \ddot{\varphi}_i \\ z_i \cos \varphi_i \ddot{\varphi}_i \end{bmatrix} = \text{accelerazione angolare}$$

$$\begin{bmatrix} -2\dot{z}_i \sin \varphi_i \dot{\varphi}_i \\ 2\dot{z}_i \cos \varphi_i \dot{\varphi}_i \end{bmatrix} = \text{accelerazione di Coriolis}$$

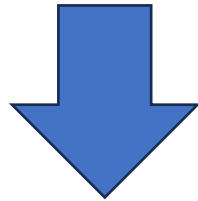
$$\begin{bmatrix} -z_i \cos \varphi_i \dot{\varphi}_i^2 \\ -z_i \sin \varphi_i \dot{\varphi}_i^2 \end{bmatrix} = \text{accelerazione centripeta}$$

```
class accelerazione(Diade):
    def __init__(self):
        self.a_co1x = -2 * Diade.z_1 * m.sin(Diade.f1) * Diade.f_1
        self.a_co1y = 2 * Diade.z_1 * m.cos(Diade.f1) * Diade.f_1
        self.a_cp1x = -Diade.z1 * m.cos(Diade.f1) * Diade.f_1**2
        self.a_cp1y = -Diade.z1 * m.sin(Diade.f1) * Diade.f_1**2
        self.a_an1x = -Diade.z1 * m.sin(Diade.f1) * Diade.f_1
        self.a_an1y = Diade.z1 * m.cos(Diade.f1) * Diade.f_1
        self.a_re1x = Diade.z__1 * m.cos(Diade.f1)
        self.a_re1y = Diade.z__1 * m.sin(Diade.f1)
```

# Analisi Accelerazione

Problema di velocità

$$\mathbf{J}\dot{\mathbf{x}} = -\mathbf{A}_q\dot{\mathbf{q}}$$



deriviamo

Problema di accelerazione

$$\mathbf{J}\ddot{\mathbf{x}} = -\mathbf{A}_q\ddot{\mathbf{q}} - \dot{\mathbf{A}}_q\dot{\mathbf{q}} - \dot{\mathbf{J}}\dot{\mathbf{x}}$$



$$\ddot{\mathbf{x}} = -\mathbf{J}^{-1} \left( \mathbf{A}_q\ddot{\mathbf{q}} + \dot{\mathbf{A}}_q\dot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{x}} \right)$$

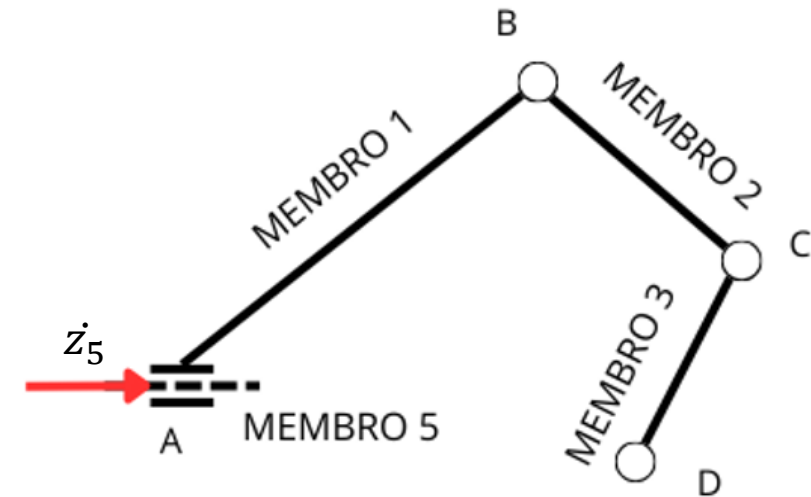
```
if(tipo_diade == "1"):
    noti = np.array([[ -self.a_cp1x - self.a_an1x - self.a_cp2x - self.a_cp3x],
                    [ -self.a_cp1y - self.a_an1y - self.a_cp2y - self.a_cp3y]])
    incognite = np.array([[self.a_an2x, self.a_an3x], [self.a_an2y, self.a_an3y]])
    risultato = np.linalg.solve(incognite, noti)
    Diade.f_2 = risultato[0][0]
    Diade.f_3 = risultato[1][0]
```

# Configurazione alternativa

## Analisi di Accelerazione e di Velocità

```
elif(ans == "2"):
    Diade.f5 = float(input("INSERIRE LA FASE DEL PATTINO DEL MECCANISMO BASE: "))
    if(tipo_diade == "1"):
        noti = np.array([[ -Diade.z_5 * m.cos(Diade.f5)], [ -Diade.z_5 * m.sin(Diade.f5)]])
        incognite = np.array([[ -Diade.z2 * m.sin(Diade.f2), -Diade.z3 * m.sin(Diade.f3)], [Diade.z2 * m.cos(Diade.f2), Diade.z3 * m.cos(Diade.f3)]])
        risultato = np.linalg.solve(incognite, noti)
        Diade.f_2 = risultato[0][0]
        Diade.f_3 = risultato[1][0]

    noti = np.array([[ -self.a_re5x], [ -self.a_re5y]])
    incognite = np.array([[self.a_an2x, self.a_an3x], [self.a_an2y, self.a_an3y]])
    risultato = np.linalg.solve(incognite, noti)
    Diade.f__2 = risultato[0][0]
    Diade.f__3 = risultato[1][0]
```





# Esempio di meccanismo risolto

## DATI

$$A = [0; 0] \text{ m}$$

$$D = [8; 3] \text{ m}$$

$$z1 = 3,5 \text{ m}$$

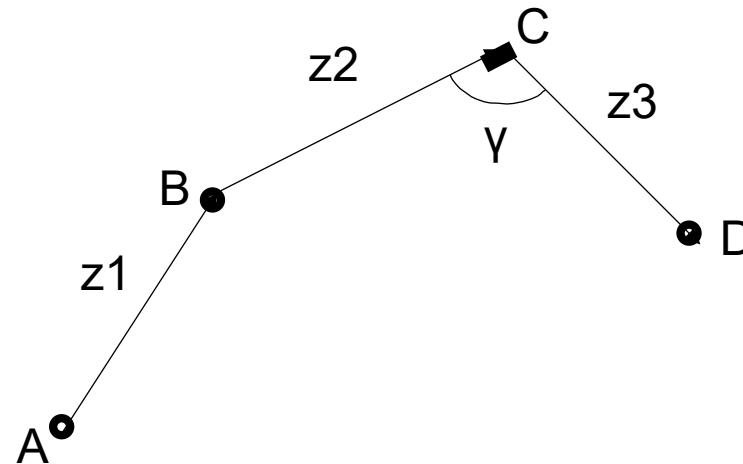
$$z3 = 2,5 \text{ m}$$

$$\gamma = 2 \text{ rad}$$

$$\varphi_1 = 1,1 \text{ rad}$$

$$\varphi'_1 = -0,5 \text{ rad/s}$$

$$\varphi''_1 = 2 \text{ rad/s}^2$$



# Esempio di meccanismo risolto

## Analisi di posizione

$$B_x = z_1 \cos(\varphi_1); B_y = z_1 \sin(\varphi_1) \rightarrow B = [1.5876; 3.1192]m$$

$$DB = \sqrt{(B_x - D_x)^2 + (B_y - D_y)^2} = 6,4135 m$$

$$\varphi_{DB} = \text{atan}\left(\frac{\Delta y}{\Delta x}\right) = 3,123 \text{ rad}$$

$$\beta = 0,3623 \text{ rad} \quad \gamma = 0,7793 \text{ rad} \quad z_2 = 4,9569 m$$

$$\varphi_2 = \varphi_{DB} - \pi + (\text{angolo in } B) = 0,3437 \text{ rad}$$

$$\varphi_3 = \varphi_2 - \pi + \alpha = -0,7979 \text{ rad}$$

# Esempio di meccanismo risolto

## Analisi di velocità

Matrice incognite: 
$$\begin{bmatrix} \cos(\varphi_2) & -z_2 * \sin(\varphi_2) - z_3 * \sin(\varphi_3) \\ \sin(\varphi_2) & z_2 * \cos(\varphi_2) + z_3 * \cos(\varphi_3) \end{bmatrix}$$

Matrice noti: 
$$\begin{bmatrix} z_1 * \sin(\varphi_1) * \varphi'_1 \\ -z_1 * \cos(\varphi_1) * \varphi'_1 \end{bmatrix}$$

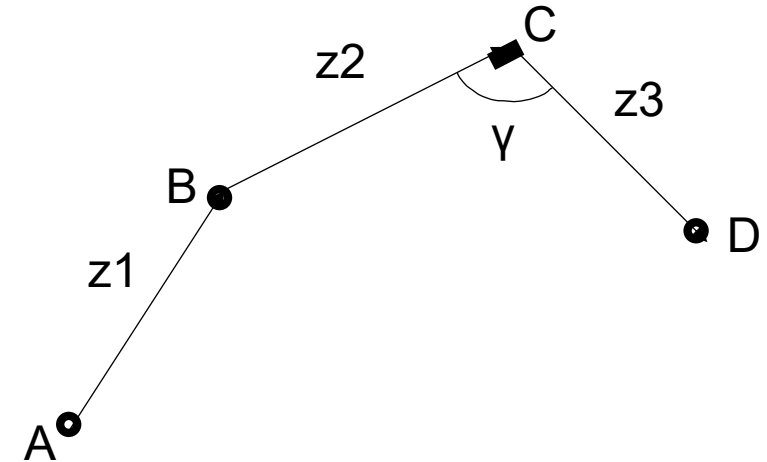
I risultati sono:  $z'_2 = -1,6586 \frac{m}{s}$      $\varphi'_2 = 0,2109 \frac{rad}{s}$

## Analisi di accelerazione

Matrice incognite: 
$$\begin{bmatrix} \cos(\varphi_2) & -z_2 * \sin(\varphi_2) - z_3 * \sin(\varphi_3) \\ \sin(\varphi_2) & z_2 * \cos(\varphi_2) + z_3 * \cos(\varphi_3) \end{bmatrix}$$

Matrice noti: 
$$\begin{bmatrix} z_1 * \cos(\varphi_1) * \varphi'^2_1 + z_1 * \sin(\varphi_1) * \varphi''_1 + \dots \\ z_1 * \sin(\varphi_1) * \varphi'^2_1 - z_1 * \cos(\varphi_1) * \varphi''_1 + \dots \end{bmatrix}$$

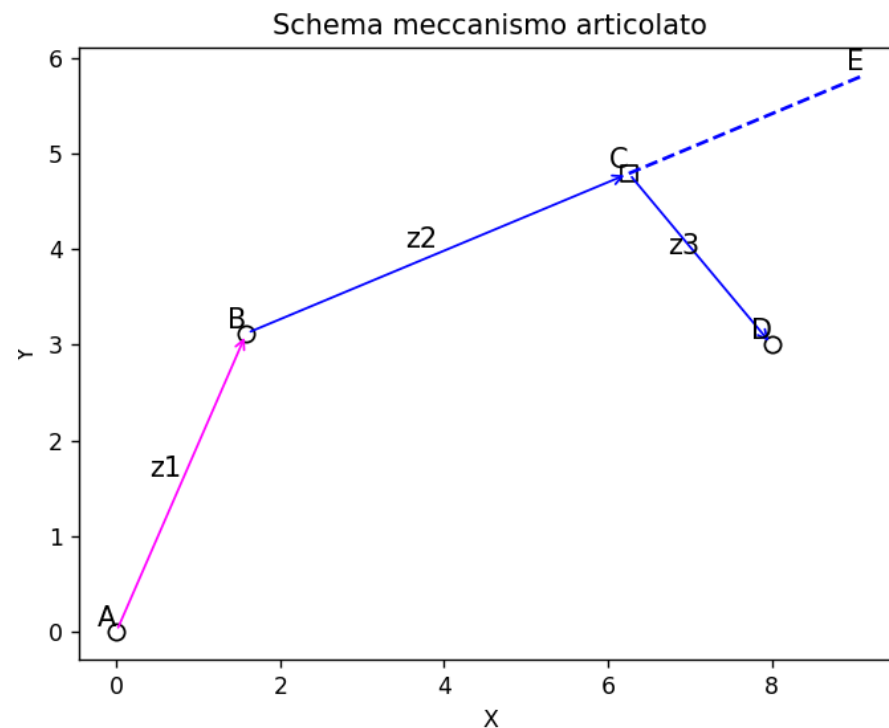
I risultati sono:  $z''_2 = 7,1825 \frac{m}{s^2}$      $\varphi''_2 = -0,6491 \frac{rad}{s^2}$



```

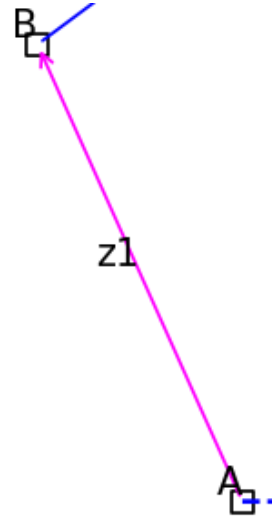
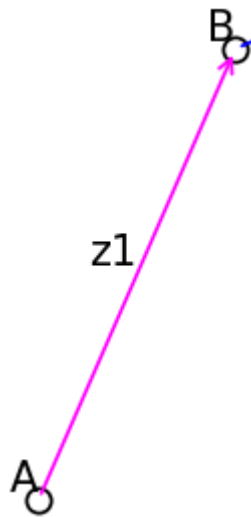
*****
MECCANISMO BASE                               INSERIRE LA COORDINATA LIBERA DI VELOCITA': -0.5
*****
COORDINATA X 1° PUNTO: 0                       VELOCITA' DIADE RPR
COORDINATA Y 1° PUNTO: 0                       Coordinata f1 punto: [-0.5000]rad/s
LUNGHEZZA 1° VETTORE: 3.5                     Coordinata f2 punto: [0.2123]rad/s
VALORE COORDINATA LIBERA f1 (IN RADIANTI): 1.1 Coordinata f3 punto: [0.2123]rad/s
*****                                         Coordinata z2 punto: [-1.6834]m/s
*****
MECCANISMO BASE                               *****
Coordinate punto A: [0.0000, 0.0000]m         INSERIRE LA COORDINATA LIBERA DI ACCELERAZIONE: 2
Coordinate punto B: [1.5876, 3.1192]m         *****
Angolo f1: [1.1000]rad                       ACCELERAZIONE DIADE RPR
Lunghezza vettore z1: [3.5000]m              Coordinata f1 doppio punto: [2.0000]rad/s^2
*****                                         Coordinata f2 doppio punto: [-0.6468]rad/s^2
DIADI RPR                                       Coordinata f3 doppio punto: [-0.6468]rad/s^2
*****                                         Coordinata z2 doppio punto: [7.1806]m/s^2
*****
COORDINATA X 2° PUNTO: 8                       *****
COORDINATA Y 2° PUNTO: 3
LUNGHEZZA 1° VETTORE: 2.5
LUNGHEZZA MEMBRO BE: 8
ANGOLO GAMMA (IN RADIANTI): 2
INDICARE LA CONFIGURAZIONE:
1)GOMITO BASSO O A SINISTRA
2)GOMITO ALTO O A DESTRA
RISPOSTA: 2
*****
DIADI RPR
Coordinata punto B: [1.5876, 3.1192]m
Coordinata punto C: [6.2544, 4.7897]m
Coordinata punto D: [8.0000, 3.0000]m
Angoli f2 e f3: [0.3437, 5.4853]rad
Lunghezza vettori z2 e z3: [4.9568, 2.5000]m
*****

```



# Print del grafico

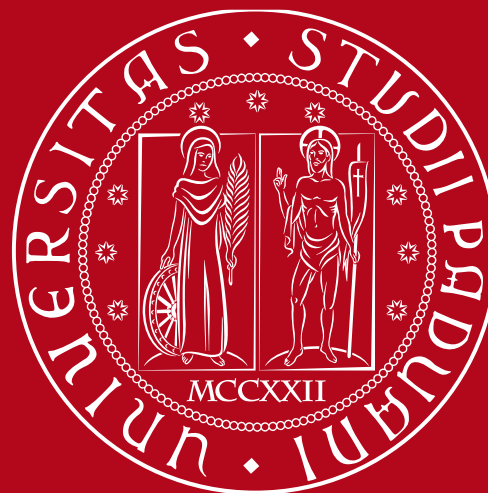
```
if(ans == "1"):  
    plt.annotate('', xy=(Diade.P2_x, Diade.P2_y), xytext=(Diade.P1_x, Diade.P1_y), arrowprops=dict(arrowstyle="->", color="magenta"))  
    plt.scatter(Diade.P1_x, Diade.P1_y, color='white', edgecolor='black', marker='o', s = 50)  
    plt.text(Diade.P1_x, Diade.P1_y, 'A', fontsize=12, ha='right', va='bottom')  
    plt.text(z1_x2, z1_y2, 'z1', fontsize=12, ha='right', va='bottom')  
elif(ans == "2"):  
    plt.annotate('', xy=(Diade.P2_x, Diade.P2_y), xytext=(Diade.P1_x, Diade.P1_y), arrowprops=dict(arrowstyle="->", color="magenta"))  
    plt.scatter(Diade.P1_x, Diade.P1_y, color='white', edgecolor='black', marker='s', s = 50)  
    plt.text(Diade.P1_x, Diade.P1_y, 'A', fontsize=12, ha='right', va='bottom')  
    plt.text(z1_x2, z1_y2, 'z1', fontsize=12, ha='right', va='bottom')  
return
```



# Bibliografia

- **Appunti dalle lezioni di Meccanica applicata alle Macchine:**  
Prof. Paolo Boscariol  
  
Immagini e formule sono state prese dalla dispensa messa a disposizione per gli studenti del corso Meccanica applicata alle macchine.
- **Logo Python:** <https://logos-world.net/python-logo/>

1222 \* 2022  
**800**  
ANNI



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA