METODI RUNGE-KUTTA

$$egin{cases} \dot{y}(t) = f(t,y(t)) \ y(0) = y_0 \end{cases}$$

$$t_0 < t_1 < ... < t_M = T$$
 , $t_{i+1} - t_i = h$

$$\int_{t_i}^{t_{i+1}} \dot{y}(t) dt = y(t_{i+1}) - y(t_i)$$

$$=\int_{t_i}^{t_{i+1}}f(t,y(t))dt$$

$$=h\int_0^1f(t_i+h_is,y(t_i+sh))ds$$

$$egin{aligned} y(t_{i+1}) \ &= y(t_i) + h \int_0^1 f(t_i + hs, y(t_i + sh)) ds \ &pprox y(t_i) + h \sum_{j=1}^s b_j f(t_i + hc_j, y(t_i + c_j h)) \end{aligned}$$

$$\int_{t_i}^{t_i+c_jh}\dot{y}(t)dt=y(t_i+c_jh)-y(t_i)$$

$$egin{aligned} \int_{t_i}^{t_i+c_jh} \dot{y}(t)dt &= y(t_i+c_jh) - y(t_i) \ &= \int_{t_i}^{t_i+c_jh} f(t,y(t))dt = \ &= h \int_0^{c_j} f(t_i+sh,y(t_i+sh))ds \ &pprox h \sum_{k=1}^s a_{jk} f(t_i+c_kh,y(t_i+c_kh)) \end{aligned}$$

Riassumendo, abbiamo ottenuto la forma di un generico metodo Runge-Kutta, che è la seguente:

$$egin{aligned} y_{i+1} &= y_i + h \sum_{j=1}^s b_j f(t_i + c_j h, Y_j) \ Y_j &= y_i + h \sum_{k=1}^s a_{jk} f(t_i + c_k h, Y_k) \end{aligned}$$

BUTCHER TABLEAU

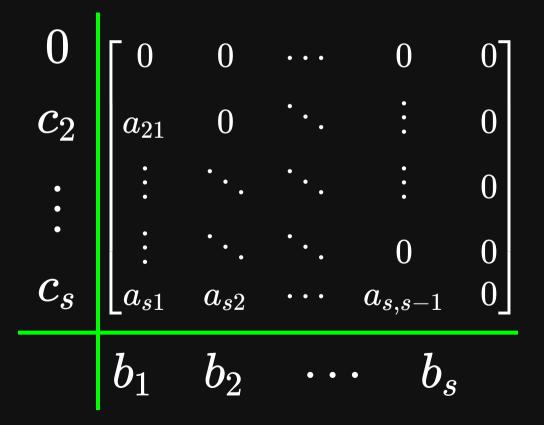
$$egin{align} y_{i+1} &= y_i + h \sum_{j=1}^s b_j f(t_i + c_j h, Y_j) \ Y_j &= y_i + h \sum_{k=1}^s a_{jk} f(t_i + c_k h, Y_k) \ \end{pmatrix}$$

PUNTO MEDIO IMPLICITO

$$egin{aligned} y_{i+1} &= y_i + h f(t_i + rac{h}{2}, Y_1) \ Y_1 &= y_i + rac{h}{2} f(t_i + rac{h}{2}, Y_1) \end{aligned}$$

$$egin{array}{c} rac{1}{2} & \left[rac{1}{2}
ight] \\ \hline 1 & \end{array}$$

RK ESPLICITO



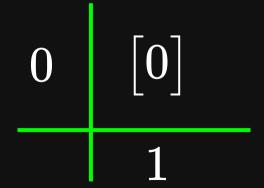
RK ESPLICITO

$$egin{aligned} y_{i+1} &= y_i + h \sum_{j=1}^s b_j f(t_i + c_j h, Y_j) \ Y_j &= y_i + h \sum_{k=1}^{j-1} a_{jk} f(t_i + c_k h, Y_k) \end{aligned}$$

$$egin{aligned} Y_1 &= y_i \ Y_2 &= y_i + ha_{21}f(t_i + c_1h, y_i) \ Y_3 &= y_i + ha_{31}f(t_i + c_1h, y_i) + \\ &+ ha_{32}f(t_i + c_2h, Y_2) \end{aligned}$$

EULERO ESPLICITO

$$egin{aligned} y_{i+1} &= y_i + h \sum_{j=1}^1 f(t_i, Y_1) \ Y_1 &= y_i \ y_{i+1} &= y_i + h f(t_i, y_i) \end{aligned}$$



```
def implicit_rk(f, time, y0, A, b, c):
    ys = np.zeros((len(y0),len(time)))
    ys[:,0] = y0
    for i in range(1, len(time)):
        h = time[i]-time[i-1]
        yVec = ys[:,i-1].repeat(len(c),axis=0)
        vec = lambda y : y.reshape(-1)
        func = lambda y : yVec + h * vec(np.einsum('ri,oi->or',A,f(y.reshape(len(y0),len(c)))) - y
        k = fsolve(func=func, x0=yVec, xtol=le-14)
        k = k.reshape((len(y0),len(c)))
        ys[:,i] = ys[:,i-1] + h * np.einsum('i,oi->o',b,f(k))
    return xs, ys
```