

FONDAMENTI DI INFORMATICA

SESTA LEZIONE, CON :
MAURO CHINAPPI
CLAUDIO VERONA

Prof. Alfredo Accattatis

(accattatis@ing.uniroma2.it)

Tutor: Prof. Vittorio Calafiore

(vcalafio@gmail.com)

Moto uniformemente accelerato

- Primo modello
- La velocità cresce indefinitamente
- Eulero con accelerazione costante e nota

Moto uniformemente accelerato

% Modello semplice, La velocità cresce indefinitamente

% Parametri del problema

a = 9.81; % Accelerazione costante (m/s²)

dt = 0.01; % Passo temporale (s)

t_max = 10.0; % Tempo finale (s)

% Inizializzazione strutture dati (vettori)

n_steps = floor(t_max / dt) + 1;

t = zeros(n_steps, 1);

y = zeros(n_steps, 1);

v = zeros(n_steps, 1);

% Condizioni iniziali

t(1) = 0.0;

y(1) = 0.0;

v(1) = 0.0;

Moto uniformemente accelerato

% Integrazione con il metodo di Eulero

for i = 2:n_steps

 t(i) = t(i-1) + dt;

 v(i) = a * t(i); % $v(t) = a * t$

 y(i) = y(i-1) + v(i-1) * dt; % $y(t+dt) = y(t) + v(t) * dt$

end

% Soluzioni analitiche

y_analitica = 0.5 * a * t.^2;

v_analitica = a * t;

Moto uniformemente accelerato

% ---

% Calcolo degli errori

% ---

err_y = abs(y_analitica - y);

err_v = abs(v_analitica - v);

Moto uniformemente accelerato

```
% ---  
% Grafico posizione  
% ---  
figure;  
plot(t(1:100:end), y(1:100:end), 'o', 'DisplayName', 'Numerical');  
hold on;  
plot(t, y_analitica, '-', 'DisplayName', 'Analytical');  
xlabel('Time (s)');  
ylabel('Position (m)');  
legend;  
grid on;
```

Moto uniformemente accelerato

```
% ---  
% Grafico velocità  
% ---  
figure;  
plot(t(1:100:end), v(1:100:end), 'o', 'DisplayName', 'Numerical');  
hold on;  
plot(t, v_analitica, '-', 'DisplayName', 'Analytical');  
xlabel('Time (s)');  
ylabel('Velocity (m/s)');  
legend;  
grid on;
```

Moto uniformemente accelerato

```
% ---
```

```
% Grafico errore posizione
```

```
% ---
```

```
figure;
```

```
subplot(2,1,1);
```

```
plot(t, err_y, 'k', 'LineWidth', 1.5);
```

```
grid on;
```

```
xlabel('Tempo [s]'); ylabel('Errore posizione [m]');
```

```
title('Errore assoluto Posizione (Analitica vs Eulero)');
```

```
subplot(2,1,2);
```

```
plot(t, err_v, 'k', 'LineWidth', 1.5);
```

```
grid on;
```

```
xlabel('Tempo [s]'); ylabel('Errore velocità [m/s]');
```

```
title('Errore assoluto Velocità (Analitica vs Eulero)');
```

Attrito lineare

Passiamo al modello più realistico

Attrito lineare

- Eulero con accelerazione calcolata in base alla resistenza

Attrito lineare

- % Parametri fisici
- $g = 9.81$; % Accelerazione gravitazionale (m/s^2)
- $r = 0.0025$; % Raggio della sfera (m)
- $\rho = 1000$; % Densità della sfera (kg/m^3)
- $\eta = 1.81\text{e-}5$; % Viscosità dinamica dell'aria ($\text{Pa}\cdot\text{s}$)
-
- % Parametri calcolati
- $\text{volume} = (4/3) * \pi * r^3$;
- $m = \rho * \text{volume}$; % Massa della sfera (kg)
- $b = 6 * \pi * \eta * r$; % Coefficiente di attrito lineare (Stokes)
-
- % Parametri numerici
- $dt = 0.01$;
- $t_{\text{max}} = 300.0$;
- $n_{\text{steps}} = \text{floor}(t_{\text{max}} / dt) + 1$;
-
- % Inizializzazione struttura dati (vettore colonna)
- $t = \text{zeros}(n_{\text{steps}}, 1)$;
- $y = \text{zeros}(n_{\text{steps}}, 1)$;
- $v = \text{zeros}(n_{\text{steps}}, 1)$;
- $a_{\text{values}} = \text{zeros}(n_{\text{steps}}, 1)$;
-
- % Condizioni iniziali
- $t(1) = 0.0$;
- $y(1) = 0.0$;
- $v(1) = 0.0$;

Attrito lineare

% Integrazione numerica (Eulero in avanti)

```
for i = 2:n_steps
    t(i) = t(i-1) + dt;
    a = (m * g - b * v(i-1)) / m;
    v(i) = v(i-1) + a * dt;
    y(i) = y(i-1) + v(i-1) * dt;
    a_values(i) = a;
end
```

% Soluzioni analitiche per confronto

```
v_terminal = m * g / b;
v_analitica = v_terminal * (1 - exp(-b * t / m));
y_analitica = v_terminal * (t + (m / b) * (exp(-b * t / m) - 1));
```

Attrito lineare

% Grafico posizione

```
figure('Position', [100, 600, 600, 400]);  
plot(t(1:500:end), y(1:500:end), 'o', 'DisplayName', 'Numerical');  
hold on;  
plot(t, y_analitica, '-', 'DisplayName', 'Analytical');  
xlabel('Time (s)');  
ylabel('Position (m)');  
legend;  
grid on;
```

% Bottone per chiusura programma

```
btn = uicontrol('Style', 'pushbutton', ...  
               'String', 'Esci', ...  
               'Position', [20, 20, 100, 40], ...  
               'Callback', @Uscita);
```

Attrito lineare

% Bottone per chiusura programma

```
btn = uicontrol('Style', 'pushbutton', ...  
               'String', 'Esci', ...  
               'Position', [20, 20, 100, 40], ...  
               'Callback', @Uscita);
```

```
function Uscita(~, ~)  
    disp('Chiusura programma');  
    close all;  
end
```

Attrito quadratico

% Integrazione numerica (Eulero in avanti)

for i = 2:n_steps

$t(i) = t(i-1) + dt;$

$F_{\text{drag}} = 0.5 * C_d * \rho_{\text{aria}} * A * v(i-1)^2;$

$F_{\text{drag}} = F_{\text{drag}} * \text{sign}(v(i-1));$

$a = (m * g - F_{\text{drag}}) / m;$

$v(i) = v(i-1) + a * dt;$

$y(i) = y(i-1) + v(i-1) * dt;$

$a_values(i) = a;$

end