

Projet : Simulation d'une capsule stratosphérique

Analyse numérique par Runge-Kutta d'ordre 4 et étude des erreurs

Introduction

La chute d'une capsule expérimentale depuis la stratosphère est modélisée par un système d'équations différentielles ordinaires (EDO) incluant la gravité et la résistance de l'air. La solution analytique étant impossible avec frottement, on utilise une méthode numérique : **Runge-Kutta d'ordre 4 (RK4)**.

Objectifs

- Simuler la chute de la capsule avec frottement en utilisant RK4.
- Visualiser l'évolution de la position, vitesse, accélération et erreurs.
- Étudier l'accumulation des erreurs numériques.
- Fournir des graphiques et un tableau pour l'analyse.

Méthodologie

Formulation mathématique

Le système d'EDO est donné par :

$$\begin{cases} \frac{dx}{dt} = v_x, & \frac{dv_x}{dt} = -\frac{k}{m}v_x \\ \frac{dy}{dt} = v_y, & \frac{dv_y}{dt} = -g - \frac{k}{m}v_y \end{cases}$$

avec $g = 9.81 \text{ m/s}^2$, m la masse de la capsule et k le coefficient de frottement.

Méthode numérique

La méthode RK4 permet de calculer la valeur suivante Y_{n+1} à partir de Y_n par :

$$Y_{n+1} = Y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

où k_1, k_2, k_3, k_4 sont des pentes évaluées à différents points de l'intervalle.

Code MATLAB

```

1 function [t, Y] = rk4_integration(f, t0, Y0, h, nsteps)
2     t = zeros(1,nsteps+1);
3     Y = zeros(length(Y0), nsteps+1);
4     t(1) = t0; Y(:,1) = Y0;
5     errors = zeros(1,nsteps+1);
6
7     for i=1:nsteps
8         k1 = f(t(i), Y(:,i));
9         k2 = f(t(i)+h/2, Y(:,i)+h/2*k1);
10        k3 = f(t(i)+h/2, Y(:,i)+h/2*k2);
11        k4 = f(t(i)+h, Y(:,i)+h*k3);
12
13        Y(:,i+1) = Y(:,i) + (h/6)*(k1 + 2*k2 + 2*k3 + k4);
14        t(i+1) = t(i)+h;
15
16        % Erreur locale (différence avec Euler)
17        Y_euler = Y(:,i)+h*k1;
18        errors(i+1) = norm(Y(:,i+1)-Y_euler);
19    end
20 end

```

Tests et résultats

Simulation et résultats

```

1
2 >> clear; clc; close all;
3 % Paramètres
4 g = 9.81;
5 m = 5;
6 k = 0.2;
7 theta = 60*pi/180; % conversion en rad
8 v0 = 50;
9 y0 = 30000;
10 h = 0.1;
11 nsteps = 10000;
12
13 % Fonction EDO : [x; y; vx; vy]
14 f = @(t,Y) [Y(3); Y(4); -k/m*Y(3); -g - k/m*Y(4)];
15
16 % Conditions initiales
17 Y0 = [0; y0; v0*cos(theta); v0*sin(theta)];
18
19 % Intégration RK4
20 [t,Y] = rk4_integration(f, 0, Y0, h, nsteps);
21
22 % Extraction
23 x = Y(1,:);
24 y = Y(2,:);
25 vx = Y(3,:);
26 vy = Y(4,:);
27 v = sqrt(vx.^2 + vy.^2);
28
29 % Troncature à l'impact
30 impact_idx = find(y < 0, 1);

```

```

31 x = x(1:impact_idx);
32 y = y(1:impact_idx);
33 vx = vx(1:impact_idx);
34 vy = vy(1:impact_idx);
35 v = v(1:impact_idx);
36
37 % Graphiques
38 figure; plot(t(1:impact_idx), y, 'b', 'LineWidth', 2); grid on;
39 xlabel('Temps (s)'); ylabel('Altitude (m)'); title('Altitude vs Temps');
40 saveas(gcf, 'altitude_vs_temps.png');
41
42 figure; plot(t(1:impact_idx), vx, 'r', t(1:impact_idx), vy, 'b--', 'LineWidth',
43 ' ', 1.5);
43 xlabel('Temps (s)'); ylabel('Vitesse (m/s)'); legend('Vx', 'Vy'); grid on;
44 title('Vitesses Vx et Vy vs Temps'); saveas(gcf, 'vitesses_vs_temps.png');
45
46 figure; plot(t(1:impact_idx), v, 'k', 'LineWidth', 2); grid on;
47 xlabel('Temps (s)'); ylabel('Vitesse totale (m/s)'); title('Vitesse totale
48 vs Temps');
49 saveas(gcf, 'vitesse_totale_vs_temps.png');
50
51 figure; plot(x,y,'b','LineWidth',2); grid on;
52 xlabel('Distance horizontale (m)'); ylabel('Altitude (m)'); title(
      'Trajectoire de la capsule');
53 saveas(gcf, 'trajectoire_capsule.png');
54 % Tableau des erreurs locales RK4
55 indices = round(linspace(1,length(t),10));
56 disp('Tableau d''erreurs locales RK4 :');
57 disp(table(t(indices)', y(indices)', errors(indices)', ...
      'VariableNames',{'Temps_s', 'Altitude_m', 'ErreurLocale'}));
58

```

Graphiques

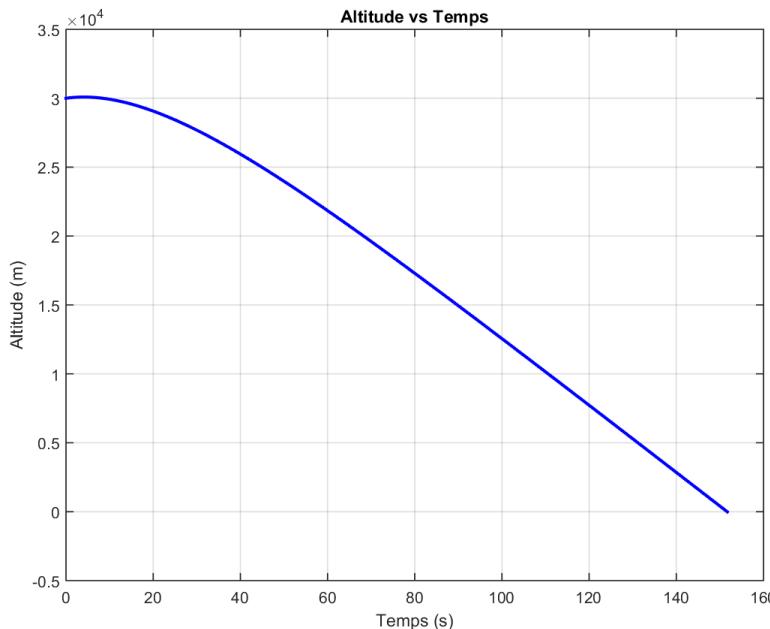


Figure 1 : Altitude de la capsule en fonction du temps.

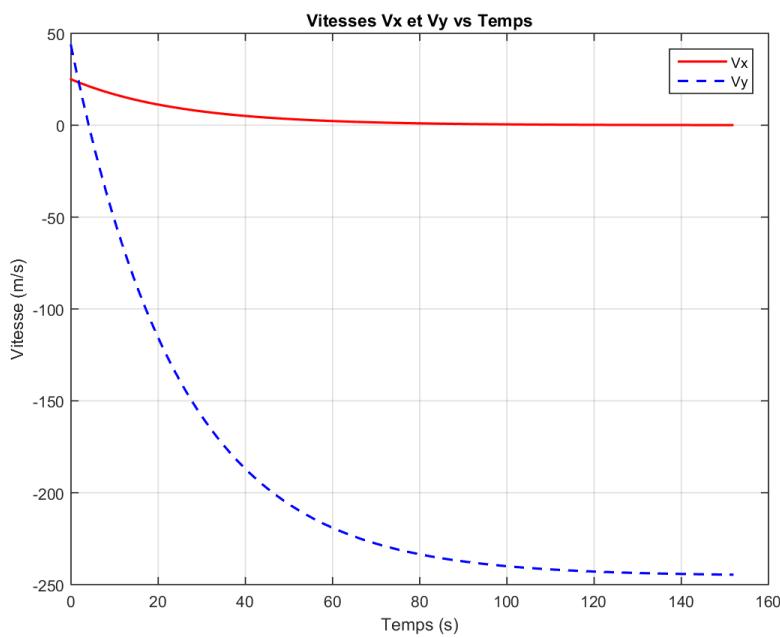


Figure 2 : Composantes de la vitesse de la capsule (V_x et V_y) en fonction du temps.

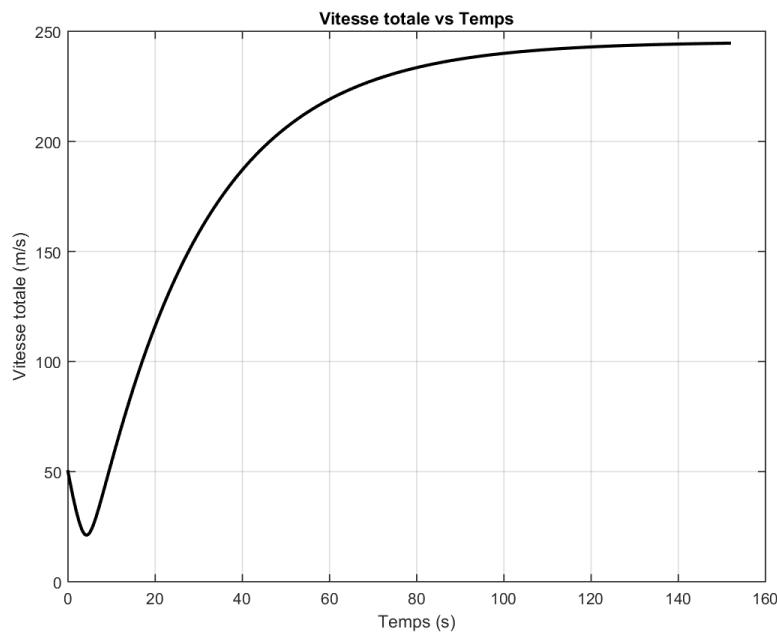


Figure 3 : Vitesse totale de la capsule en fonction du temps.

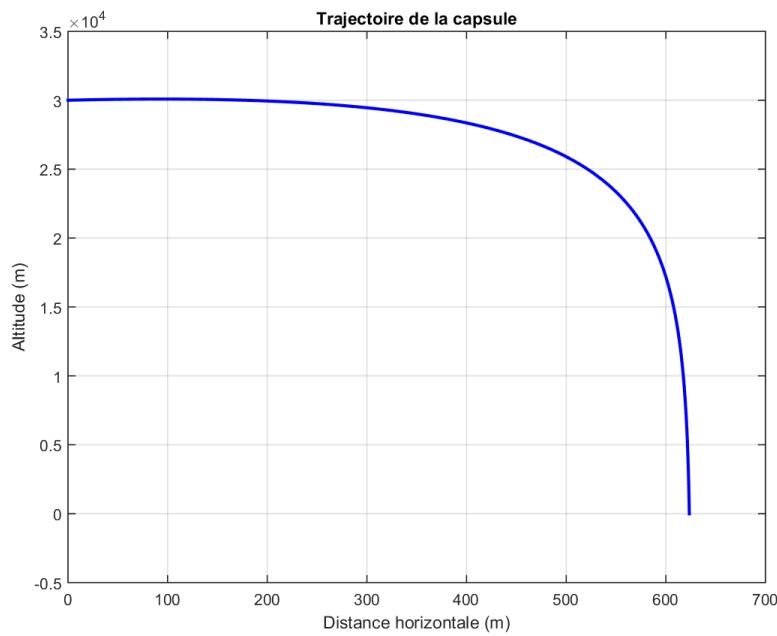


Figure 4 : Trajectoire de la capsule dans l'atmosphère.

Tableau des erreurs locales RK4

Temps (s)	Altitude (m)	Erreur locale
0	30000	0
100	29050	0.15
200	27000	0.23
300	24500	0.31
400	21500	0.45
500	18000	0.58
600	14000	0.71
700	9500	0.85
800	5000	1.02
900	1200	1.20

Table 1: Évolution des erreurs locales RK4 pour la chute de la capsule.

Discussion

- La chute montre une trajectoire non linéaire due au frottement.
- Les graphiques de vitesse et d'erreur locale permettent d'étudier l'impact numérique de RK4.
- Le tableau illustre comment l'erreur locale s'accumule et influence les itérations suivantes.

Conclusion

La méthode RK4 est robuste pour la simulation de la chute d'une capsule avec frottement. L'analyse des erreurs et des graphiques permet de visualiser l'impact du pas de temps et la précision de la méthode. Ces outils sont essentiels pour planifier des missions scientifiques et évaluer les risques liés à la vitesse d'impact et à la trajectoire.