Numerical solution of ODEs
Some of the numerical solving algorithms for ODEs

Numerical solutions are approximate solutions to the differential equation where the analytical solution is hard to get or even not available as the case in airplane equations. The variation of the algorithms is due to the need for different approaches for solving the equations like: Accuracy requirements, stability considerations, computational efficiency, step size adaptability, and system constraints.

Reason for the Variation of Numerical Algorithms. The wide variety of numerical algorithms for solving Ordinary Differential Equations (ODEs) exists because different problems have unique characteristics that require specialized approaches. The main factors influencing algorithm selection include:

1. Accuracy Requirements – Some applications demand high precision (e.g., spacecraft navigation), while others prioritize speed (e.g., real-time control systems).

2. Stability Considerations – Certain ODEs, especially stiff systems (e.g., chemical reactions, flight dynamics at high speeds), require implicit methods for stable solutions.

3. Computational Efficiency – Simpler methods (e.g., Euler's) require fewer calculations but may need smaller time steps, while higher-order methods (e.g., Runge-Kutta) balance accuracy and efficiency.

4. Step Size Adaptability – Some methods (e.g., RK45) adjust the step size automatically to handle rapidly changing or smooth regions efficiently.

5. System Constraints – Real-time applications (e.g., autopilot, robotics) need algorithms that work under computational and time constraints.

Benefits of Algorithm Variation

1. Optimized Performance for Different Applications
- Euler's Method is useful for quick approximations.
- RK4 provides high accuracy for aerospace simulations.
- Implicit methods (e.g., Crank-Nicolson) are better for stiff problems like structural vibrations.

2. Balancing Accuracy & Speed
- High-order methods (RK4, RK45) are more precise but computationally expensive.
- Lower-order methods (Euler, Adams-Bashforth) are faster but may require smaller steps.

3. Handling Stiff vs. Non-Stiff Systems
- Stiff ODEs (e.g., high-speed aerodynamic models) require implicit methods for stability.
- Non-stiff ODEs (e.g., basic motion equations) can use explicit methods.

4. Adaptability to Problem Complexity
- Verlet integration conserves energy in orbital mechanics.
- Predictor-Corrector methods improve accuracy in long-duration flight simulations.

5. Real-Time Applications
- Some algorithms (e.g., fixed-step RK4) are used in autopilot systems to meet real-time constraints.
- Adaptive methods (RK45) are useful in simulations but may be too slow for real-time control

Summary of Numerical Algorithms for Solving ODEs

Method	Order	Type	Pros	Cons
Euler's Method	1st	Explicit	Simple, fast	Low accuracy, unstable for large steps
RK4	4th	Explicit	High accuracy, widely used	Computationally expensive
RK45	Adaptive	Explicit	Efficient, variable step size	More complex
Adams-Bashforth	Variable	Explicit	Efficient, reuses previous values	Requires initialization
Adams-Moulton	Variable	Implicit	Stable, good for stiff problems	Computational overhead
Backward Euler	1st	Implicit	Stable for stiff ODEs	Requires solving nonlinear equations
Crank-Nicolson	2nd	Implicit	Stable, accurate	More complex than Backward Euler
Verlet Integration	2nd	Explicit	Energy-conserving	Requires two previous states

- Explicit Methods: Easier to implement but may require small step sizes for stability.
- Implicit Methods: More stable, especially for stiff problems, but require solving equations at each step.
- Adaptive Methods: Adjust step size automatically for better efficiency.
- Specialized Methods: Used for specific applications like orbital mechanics (Verlet) or flight control (Predictor-Corrector).

Choosing one algorithm for solving the Airplanes EOM, clearly state the (Initial conditions needed, Inputs needed in each iteration, and Outputs calculated in each iteration)

The equation of motion of an airplane is derived from the rigid body motion equations. Those equations could be divided into five (or four) sets of equations:

1. Force equations

$$X - mgS_\theta = m(\dot{u} + qw - rv)$$

$$Y + mgC_\theta S_\Phi = m(\dot{v} + ru - pw)$$

$$Z + mgC_\theta C_\Phi = m(\dot{w} + pv - qu)$$

2. Moment equations

$$L = I_x\dot{p} - I_{xz}\dot{r} + qr(I_z - I_y) - I_{xz}pq$$

$$M = I_y\dot{q} + rq(I_x - I_z) + I_{xz}(p^2 - r^2)$$

$$N = -I_{xz}\dot{p} + I_z\dot{r} + pq(I_y - I_x) + I_{xz}qr$$

3. Angular velocity

$$p = \dot{\Phi} - \dot{\psi}S_\theta$$

$$q = \dot{\theta}C_\Phi + \dot{\psi}C_\theta S_\Phi$$

$$r = \dot{\psi}C_\theta C_\Phi - \dot{\theta}S_\Phi$$

4. Velocity equations

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dx}{dt} \\ \frac{dz}{dt} \end{bmatrix} = \begin{bmatrix} C_\theta C_\psi & S_\Phi S_\theta C_\psi - C_\Phi S_\psi & C_\psi S_\theta C_\psi + S_\Phi S_\psi \\ C_\theta S_\psi & S_\Phi S_\theta S_\psi + C_\Phi C_\psi & C_\Phi S_\theta S_\psi - S_\Phi C_\psi \\ -S_\theta & S_\Phi C_\theta & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

We need to get every derivative W.R.T. time so we will apply the small disturbance approximation where we could swap every variable with an initial known value and the change in this value. This approximation is only valid when the changes in those variables (states) are very small which could be seen almost at the cruise conditions throughout the journey of an aircraft.

The variables could be replaced as follows:

$$\boldsymbol{u = u_0 + \Delta u}$$
$$\boldsymbol{v = v_0 + \Delta v}$$
$$\boldsymbol{w = w_0 + \Delta w}$$
$$\boldsymbol{p = p_0 + \Delta p}$$
$$\boldsymbol{q = q_0 + \Delta q}$$
$$\boldsymbol{r = r_0 + \Delta r}$$
$$\boldsymbol{X = X_0 + \Delta X}$$
$$\boldsymbol{Y = Y_0 + \Delta Y}$$
$$\boldsymbol{Z = Z_0 + \Delta Z}$$
$$\boldsymbol{M = M_0 + \Delta M}$$
$$\boldsymbol{N = N_0 + \Delta N}$$
$$\boldsymbol{L = L_0 + \Delta L}$$
$$\boldsymbol{\delta = \delta_0 + \Delta \delta}$$

Flight conditions are taken to be symmetric to simplify the equation to be:

$$\left.\begin{aligned}
\boldsymbol{\Delta X} &= \boldsymbol{\frac{\partial X}{\partial u}\Delta u + \frac{\partial X}{\partial w}\Delta w} \\
\boldsymbol{\Delta Y} &= \boldsymbol{\frac{\partial Y}{\partial v}\Delta v + \frac{\partial Y}{\partial p}\Delta p + \frac{\partial Y}{\partial r}\Delta r} \\
\boldsymbol{\Delta Z} &= \boldsymbol{\frac{\partial Z}{\partial u}\Delta u + \frac{\partial Z}{\partial w}\Delta w + \frac{\partial Z}{\partial \dot{w}}\Delta \dot{w} + \frac{\partial Z}{\partial q}\Delta q}
\end{aligned}\right\}$$

$$\boldsymbol{\Delta L = \frac{\partial L}{\partial v}\Delta v + \frac{\partial L}{\partial p}\Delta p + \frac{\partial L}{\partial r}\Delta r}$$
$$\boldsymbol{\Delta M = \frac{\partial M}{\partial u}\Delta u + \frac{\partial M}{\partial w}\Delta w + \frac{\partial M}{\partial \dot{w}}\Delta \dot{w} + \frac{\partial M}{\partial q}\Delta q}$$
$$\boldsymbol{\Delta N = \frac{\partial N}{\partial v}\Delta v + \frac{\partial N}{\partial p}\Delta p + \frac{\partial N}{\partial r}\Delta r}$$

We will choose RK4 as our algorithm for solving the EOM, and here are the ICs, BCs, and the outputs of each iteration:

1. Initial conditions - Before starting the numerical integration, we need the initial state of the aircraft, which typically includes:
   - Position: (initial location in a given coordinate system)
   - Velocity: (initial velocity components in body frame)
   - Attitude (Orientation): (initial roll, pitch, and yaw angles)
   - Angular Velocity: (initial roll, pitch, and yaw rates)

- Mass & Inertia Properties: (aircraft mass and moments of inertia)

2. Input needed in each iteration - At each time step, the numerical algorithm needs updated information to compute the next state:

Aerodynamic Forces & Moments:

Lift, Drag, and Side Forces
Moments about body axes

Control Inputs (from Pilot or Autopilot):
Throttle Setting
Elevator Deflection
Aileron Deflection
Rudder Deflection

Environmental Conditions:
Wind Velocity Components
Air Density
Gravity

3. Outputs Calculated in Each Iteration -The numerical algorithm updates and computes the next state of the aircraft:
    - Updated position
    - Updated velocity components
    - Updated attitude (orientation)
    - Updated angular velocity
    - New forces & moments