Heun's Method and its Applications

Numerical Methods for Dynamical Systems

Ratthaprom PROMKAM, Dr. rer. nat.

Department of Mathematics and Computer Science, RMUTT

Heun's method (modified Euler's method) is a predictor-corrector technique for solving IVP that improves Euler's method by reducing errors and increasing accuracy. It estimates the solution using an initial prediction and refines it with an average slope.

Initial Value Problem (IVP)

$$\dot{x}(t) = f\big(t, x(t)\big), \quad x(t_0) = x_0.$$

Key Idea: To achieve a more accurate solution, we compute the slope at the current step and then adjust it at the next step using the following equations.

Heun's Method

Step 1: Predictor (Euler's Estimate) We first estimate the next value using Euler's method:

$$x^0(t+h) \approx x(t) + hf(t, x(t)).$$

This estimated value is called the predictor.

Step 2: Corrector (Refined Estimate) Next, we use the predictor value to compute an updated slope and refine our estimation:

$$x(t+h) \approx x(t) + h\frac{f(t, x(t)) + f(t+h, x^0(t+h))}{2}.$$

This correction improves the accuracy compared to Euler's method.

Heun's Method

Step 3: Discretize the Time Domain

- ✎ Define $t_n = t_0 + nh$, where h is the step size.
- ✎ Let $x_n \approx x(t_n)$.

Step 4: Derive the Update Formula

Applying the approximation:

$$x_{n+1}^0 = x_n + h\,f\big(t_n, x_n\big).$$
$$x_{n+1} = x_n + h\frac{f\big(t_n, x_n\big) + f\big(t_{n+1}, x_{n+1}^0\big)}{2}$$

Heun's Method

Heun Algorithmic Implementation

1. **Initialize:** x_0, t_0, h.
2. **Loop:**

$$\begin{aligned}
\text{Predictor:} \quad & x_{n+1}^0 = x_n + h\, f(t_n, x_n), \\
\text{Corrector:} \quad & x_{n+1} = x_n + h\frac{f\big(t_n, x_n\big) + f\big(t_{n+1}, x_{n+1}^0\big)}{2} \\
& t_{n+1} = t_n + h.
\end{aligned}$$

3. **Stop:** when t_n reaches (or exceeds) final time T.

Heun's Method

Local Truncation Error (LTE): The local truncation error of Heun's method is given by:

$$x(t+h) - \left(x(t) + \frac{h}{2} \left[f(t, x(t)) + f(t+h, x^0(x+h)) \right] \right) = \mathcal{O}(h^3).$$

This improves the error from $\mathcal{O}(h^2)$ in Euler's method to $\mathcal{O}(h^3)$.

Global Error:

- ✎ Over n steps, the total (global) error is $\mathcal{O}(h^2)$.

Summary:

- ✎ Heun's method provides better accuracy than Euler's method by correcting the initial estimate.
- ✎ Accuracy improves significantly while maintaining a relatively simple implementation.

Example (Exponential Growth)

Solve numerically using Heun's method:

$$\dot{x} = x, \quad x(0) = 1, \quad h = 0.1.$$

Solution: Using Heun's method:

$$x_{n+1}^0 = x_n + h\, x_n.$$

$$x_{n+1} = x_n + h\frac{x_n + x_{n+1}^0}{2}$$

Starting at $x_0 = 1$:

$$x_1^0 = 1 + 0.1(1) = 1.1.$$

$$x_1 = 1 + (0.1)(\frac{1 + 1.1}{2}) = 1.105.$$

$$x_2^0 = 1.105 + 0.1(1.105) = 1.2155.$$

$$x_2 = 1.105 + (0.1)(\frac{1.105 + 1.2155}{2}) = 1.221.$$

Example (Exponential Growth)

Solve numerically using Heun's method:

$$\dot{x} = x, \quad x(0) = 1, \quad h = 0.1.$$

$$x_3^0 = 1.221 + 0.1(1.221) = 1.3431.$$
$$x_3 = 1.221 + (0.1)(\frac{1.221 + 1.3431}{2}) = 1.3492.$$

Compare with exact solution: $x(t) = e^t$.
Observation: Heun's method slightly underestimates the true solution but has better accuracy than Euler's method.

Example

Solve numerically using Heun's method:

$$\dot{x} = x\cos(t), \quad x(0) = 1, \quad h = 1.$$

Solution: Using Heun's method:

$$x_{n+1}^0 = x_n + h\,(x_n\cos(t_n)).$$
$$x_{n+1} = x_n + h\frac{(x_n\cos(t_n)) + (x_{n+1}\cos(t_{n+1}))}{2}$$

Starting at $x_0 = 1, t_0 = 0$ and $h = 1$:

$$x_1^0 = 1 + (1)\cdot((1)\cos(0)) = 2.$$
$$x_1 = 1 + (1)\cdot\frac{((1)\cos(0)) + ((2)\cos(1))}{2}$$
$$= 2.0403$$

Example

Solve numerically using Heun's method:

$$\dot{x} = x\cos(t), \quad x(0) = 1, \quad h = 1.$$

$$x_2^0 = 2.0403 + (1)\cdot((2.0403)\cos(1)) = 3.1427.$$

$$\begin{aligned} x_2 &= 2.0403 + (1)\cdot\frac{((2.0403)\cos(1)) + ((3.1427)\cos(2))}{2} \\ &= 1.9376 \end{aligned}$$

Compare with exact solution: $x(t) = e^{\sin(t)}$.
Observation: Heun's method provides reasonable accuracy for small t but deviates for larger t.

Example (Simple Harmonic Oscillator)

Consider the second-order ODE:

$$\ddot{x} + x = 0, \quad x(0) = 1, \quad \dot{x}(0) = 0.$$

Step 1: Convert to a First-Order System in Vector Form

Let

$$\mathbf{s} = \begin{bmatrix} x \\ y \end{bmatrix} \quad \text{where } y = \dot{x}.$$

Then,

$$\dot{\mathbf{s}}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} y(t) \\ -x(t) \end{bmatrix} =: \mathbf{f}\big(\mathbf{y}(t)\big).$$

Example (Simple Harmonic Oscillator)

Consider the second-order ODE:

$$\ddot{x} + x = 0, \quad x(0) = 1, \quad \dot{x}(0) = 0.$$

Step 2: Apply Heun's Method

$$Predictor : \mathbf{s}_{n+1}^{0} = \mathbf{s}_n + h\,\mathbf{f}(\mathbf{s}_n).$$

and

$$Corrector : \mathbf{s}_{n+1} = \mathbf{s}_n + h\,\frac{\mathbf{f}(\mathbf{s}_n) + \mathbf{f}(\mathbf{s}_{n+1}^{0})}{2}.$$

Example (Simple Harmonic Oscillator)

Consider the second-order ODE:

$$\ddot{x} + x = 0, \quad x(0) = 1, \quad \dot{x}(0) = 0.$$

In coordinates, this becomes:

$$\begin{bmatrix} x_{n+1}^0 \\ y_{n+1}^0 \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} + h \begin{bmatrix} y_n \\ -x_n \end{bmatrix} = \begin{bmatrix} x_n + h\, y_n \\ y_n - h\, x_n \end{bmatrix}.$$

and

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} + h \begin{bmatrix} \frac{y_n + y_{n+1}^0}{2} \\ -\frac{x_n + x_{n+1}^0}{2} \end{bmatrix} = \begin{bmatrix} x_n + h\, \frac{y_n + y_{n+1}^0}{2} \\ y_n - h\, \frac{x_n + x_{n+1}^0}{2} \end{bmatrix}.$$

Example (Simple Harmonic Oscillator)

Consider the second-order ODE:

$$\ddot{x} + x = 0, \quad x(0) = 1, \quad \dot{x}(0) = 0.$$

Step 3: Implementation Example

Initial conditions: $\mathbf{s}_0 = \mathbf{s}(0) = \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad h = 0.1.$

$$\mathbf{s}_1^0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.1 \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ -0.1 \end{bmatrix}.$$

$$\mathbf{s}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.1 \begin{bmatrix} \frac{0+(-0.1)}{2} \\ -\frac{1+1}{2} \end{bmatrix} = \begin{bmatrix} 1 - 0.005 \\ 0 - 0.1 \end{bmatrix} = \begin{bmatrix} 0.995 \\ -0.1 \end{bmatrix}.$$

Example (Simple Harmonic Oscillator)

Consider the second-order ODE:

$$\ddot{x} + x = 0, \quad x(0) = 1, \quad \dot{x}(0) = 0.$$

$$\mathbf{s}_2^0 = \begin{bmatrix} 0.995 \\ -0.1 \end{bmatrix} + 0.1 \begin{bmatrix} -0.1 \\ -0.995 \end{bmatrix} = \begin{bmatrix} 0.985 \\ -0.1995 \end{bmatrix}.$$

$$\begin{aligned} \mathbf{s}_2 &= \begin{bmatrix} 0.995 \\ -0.1 \end{bmatrix} + 0.1 \begin{bmatrix} \dfrac{-0.1 + (-0.1995)}{2} \\ -\dfrac{0.995 + 0.985}{2} \end{bmatrix} \\ &= \begin{bmatrix} 0.995 - 0.015 \\ -0.1 - 0.099 \end{bmatrix} = \begin{bmatrix} 0.98 \\ -0.199 \end{bmatrix}. \end{aligned}$$

Example (Simple Harmonic Oscillator)

Consider the second-order ODE:

$$\ddot{x} + x = 0, \quad x(0) = 1, \quad \dot{x}(0) = 0.$$

Compare with exact solution: $x(t) = \cos(t)$.

Observation: Heun's method significantly reduces numerical damping and energy drift for oscillatory systems. Compared to Euler's method, it provides a more accurate approximation of the oscillatory behavior and better preserves the amplitude over time. However, for long-term simulations, small numerical dissipation may still occur, leading to slight deviations from the exact solution.