

Course 02/1

Ordinary differential equations (2/2)

- Predictor-corrector method
- Runge-Kutta method
- Some stability considerations

Predictor-corrector method

Through an example: the harmonic oscillator

$$\dot{p} = f(x)$$

$$\dot{x} = p / m$$

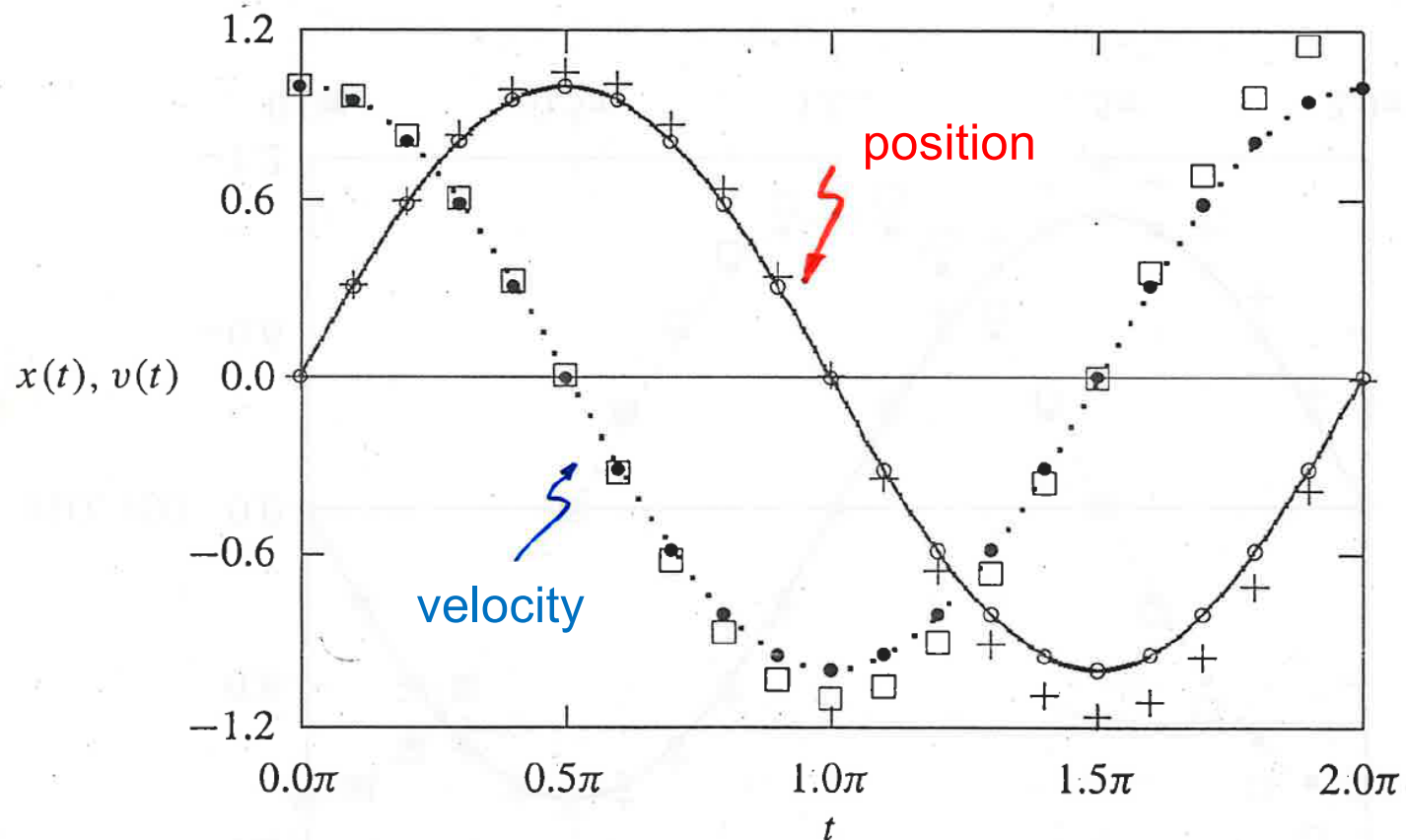
1. Predictor step (Euler)

$$x(t + \Delta t) = x(t) + \Delta t \dot{x}(t) = x(t) + \Delta t p(t) / m + o(\Delta t^2)$$

$$p(t + \Delta t) = p(t) + \Delta t \dot{p}(t) = p(t) + \Delta t f(x(t)) + o(\Delta t^2)$$

Example of predictor-corrector

Harmonic oscillator



Euler : +, □

predictor-corrector : ○, ●

Predictor-corrector method

2. Corrector step (implicit method)

$$x(t + \Delta t) = x(t) + \int_t^{t+\Delta t} p(t')/m \, dt' + o(\Delta t^3)$$

$$p(t + \Delta t) = p(t) + \int_t^{t+\Delta t} f(t') \, dt' + o(\Delta t^3)$$

$$x(t + \Delta t) = x(t) + \Delta t \left(\frac{p(t) + p(t + \Delta t)}{2m} \right) + o(\Delta t^3)$$

$$p(t + \Delta t) = p(t) + \Delta t \left(\frac{f(t) + f(t + \Delta t)}{2} \right) + o(\Delta t^3)$$

Estimate provided through Euler method
(predictor step)

Predictor-corrector scheme

1. Predictor step (Euler)

$$x^P(t + \Delta t) = x(t) + \Delta t \, p(t) / m + o(\Delta t^2)$$

$$p^P(t + \Delta t) = p(t) + \Delta t \, f(t) + o(\Delta t^2)$$

2. Corrector step

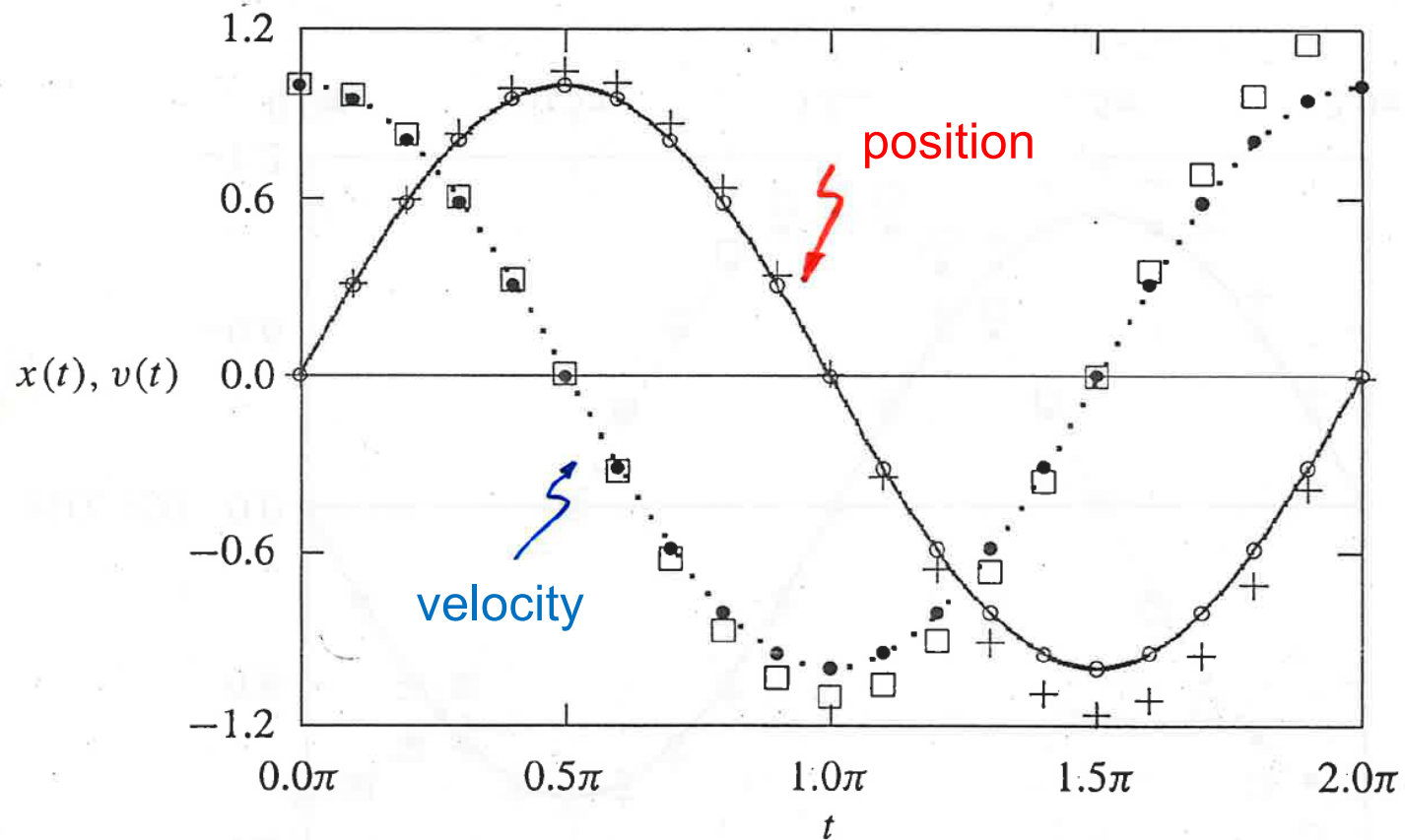
New “force” evaluation $f(t + \Delta t) = f(x^P)$

$$\begin{aligned} x^C(t + \Delta t) &= x(t) + \Delta t \left(\frac{p(t) + p^P(t + \Delta t)}{2m} \right) + o(\Delta t^3) \\ &= x^P(t + \Delta t) + \Delta t \left(\frac{p^P(t + \Delta t) - p(t)}{2m} \right) + o(\Delta t^3) \end{aligned}$$

$$\begin{aligned} p^C(t + \Delta t) &= p(t) + \Delta t \left(\frac{f(t) + f(x^P)}{2} \right) + o(\Delta t^3) \\ &= p^P(t) + \Delta t \left(\frac{f(x^P) - f(t)}{2} \right) + o(\Delta t^3) \end{aligned}$$

Example of predictor-corrector

Harmonic oscillator



Euler : +, □

predictor-corrector : ○, ●

Predictor-corrector code for harmonic oscillator

```
PROGRAM ONE_D_MOTION2
C
C Simplest predictor-corrector algorithm applied to
C a particle in one dimension under an elastic force.
C
      PARAMETER (N=101,IN=5)
      REAL T(N),V(N),X(N)
      PI = 4.0*ATAN(1.0)
      DT = 2.0*PI/100
      X(1)= 0.0
      T(1)= 0.0
      V(1)= 1.0
      DO      100   I = 1, N-1
          T(I+1) = I*DT
C
C Predictor for position and velocity
C
          X(I+1) = X(I)+V(I)*DT
          V(I+1) = V(I)-X(I)*DT
C
C Corrector for position and velocity
C
          X(I+1) = X(I)+(V(I)+V(I+1))*DT/2.0
          V(I+1) = V(I)-(X(I)+X(I+1))*DT/2.0
100 CONTINUE
      WRITE (6,999) (T(I),X(I),V(I),I=1,N,IN)
      STOP
999 FORMAT (3F16.8)
      END
```

Runga-Kutta method

The method can be developed at different orders. Here, we illustrate the 2nd order for simplicity.

Implicit method $y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x, y) dx$

1. Approximation of the integral

$$f(x, y) = f(x_{n+1/2}, y_{n+1/2}) + (x - x_{n+1/2}) f'_{n+1/2} + \dots$$

$$\int_{x_n}^{x_{n+1}} f(x, y) dx = h f(x_{n+1/2}, y_{n+1/2}) + \text{---} + o(h^3)$$

we obtain

$$y_{n+1} = y_n + h f(x_{n+\frac{1}{2}}, y_{n+\frac{1}{2}}) + o(h^3)$$

Runga-Kutta method

$$y_{n+1} = y_n + h f(x_{n+1/2}, y_{n+1/2}) + o(h^3)$$

2. Euler for $f(x_{n+1/2}, y_{n+1/2})$

Approximation to order $o(h^2)$ is sufficient

$$y_{n+1/2} = y_n + \frac{1}{2} h f(x_n, y_n) + o(h^2)$$

We define

$$k = h f(x_n, y_n)$$

By combining we get

$$y_{n+1} = y_n + h f(x_n + h/2, y_n + k/2) + o(h^3)$$

Runga-Kutta method: advantages vs disadvantages

$$\left\{ \begin{array}{l} k = h f(x_n, y_n) \\ y_{n+1} = y_n + h f(x_n + h/2, y_n + k/2) + o(h^3) \end{array} \right.$$

Disadvantage: For every step, the function f is calculated twice.
Once for (x_n, y_n) and once for $(x_n + h/2, y_n + k/2)$.
In 4th order Runge-Kutta, the function is calculated four times.

Advantages:

1. Accuracy of order $o(h^3)$.
2. No condition on easy differentiability (cf. Taylor).
3. Can start right away (cf. Multistep).
4. Does not require linearity of function f (cf. Implicit).
5. Size of "time" step can be changed at any time.

Some stability considerations

The stability of an algorithm is an important issue, it concerns the extent to which the noise due to the round-off errors is amplified.

To illustrate this we compare the Euler method (two-point formula) with a scheme that is derived from the derivative established through symmetric difference (three-point formula):

$$y_{n+1} = y_n + h f(x_n, y_n) + o(h^2)$$

$$y_{n+1} = y_{n-1} + 2h f(x_n, y_n) + o(h^3)$$

Case study:

$$f(x, y) = \frac{dy}{dx} = -y \qquad y(x=0) = 1$$

Exact analytical solution: $y = e^{-x}$

Some stability considerations

Case study:

$$f(x, y) = \frac{dy}{dx} = -y \qquad y(x=0) = 1$$

Numerical solution

Start with Taylor method:

$$y_{n+1} = y_n + h f + \frac{1}{2} h^2 \left[\frac{\partial f}{\partial x} + f \frac{\partial f}{\partial y} \right] + o(h^3)$$

$$y_1 = 1 - h + \frac{1}{2} h^2 + o(h^3)$$

Recursion formula with symmetric difference:

$$y_{n+1} = y_{n-1} - 2 h y_n + o(h^3)$$

Some stability considerations: numerical evolution

Exact solution: $y = e^{-x}$

Table 2.2 Integration of $dy/dx = -y$ with $y(0)=1$ using Eq. (2.27)

x	Exact	Error	x	Exact	Error	x	Exact	Error
0.2	.818731	-.000269	3.3	.036883	-.000369	5.5	.004087	-.001533
0.3	.740818	-.000382	3.4	.033373	-.000005	5.6	.003698	.001618
0.4	.670320	-.000440	3.5	.030197	-.000380	5.7	.003346	-.001858
0.5	.606531	-.000517	3.6	.027324	.000061	5.8	.003028	.001989
0.6	.548812	-.000538	3.7	.024724	-.000400	5.9	.002739	-.002257
			3.8	.022371	.000133	6.0	.002479	.002439

- For large x , the error increases
- The error changes sign at every step

How can we understand this?

Case study:

$$f(x, y) = \frac{dy}{dx} = -y \qquad y(x=0) = 1$$

Ansatz for numerical solution: $y_n = A r^n$

$$A r^{n+1} = A r^{n-1} - 2 h A r^n$$

$$r^2 = 1 - 2 h r \quad \Rightarrow \quad r^2 + 2 h r - 1 = 0$$

$$r_{\pm} = -h \pm \sqrt{h^2 + 1}$$

$$\cong -h \pm (1 + \tfrac{1}{2} h^2 + \dots)$$

$$h \ll 1$$

$$\cong \pm 1 - h \quad = \begin{cases} 1 - h \\ -(1 + h) \end{cases}$$

How can we understand this?

General solution:

$$y_n \cong \underbrace{A(1-h)^n}_{\text{vanishing solution (regular)}} + \underbrace{B(-1)^n(1+h)^n}_{\text{diverging solution (spurious)}}$$

- In a numerical solution, the coefficients A and B are going to be determined by the choice of y_0 and y_1 . Since y_1 is approximate, $B \neq 0$ and the spurious solution will eventually dominate.
- Even if we would start with the exact y_1 , the spurious solution would eventually creep in because of the round-off errors.

Course 02/1

Ordinary differential equations (2/2)

- Predictor-corrector method
- Runge-Kutta method
- Some stability considerations