```
> # Define the Taylor expansion for a forward approximation
Taylor_y := (h, t) → y(t) + h*diff(y(t), t) + (1/2)*h^2*diff(y(t), t$2) + (1/6)
    *h^3*diff(y(t), t$3) + (1/24)*h^4*diff(y(t), t$4) :

# Define the Taylor expansion for a backward approximation
Taylor_g := (h, t) → y(t) − h*diff(y(t), t) + (1/2)*h^2*diff(y(t), t$2) − (1/6)
    *h^3*diff(y(t), t$3) + (1/24)*h^4*diff(y(t), t$4) :

# Define the result expression
result := (h, t) → (Taylor_y(h, t) + Taylor_g(h, t) − 2*y(t)) / h^2 :

# Simplify the result
simplified_result := simplify(result(h, t));
```

$$simplified\_result := \frac{h^2 \left(\frac{d^4}{dt^4}\, y(t)\right)}{12} + \frac{d^2}{dt^2}\, y(t) \qquad \textbf{(1)}$$

```
>
#i) kwadratisch
> #ii
> restart;
with(plots) :
with(plottools) :
```

$t\_val := evalf\left(\frac{\text{Pi}}{3}\right);$

```
# Define the function y and its 2nd derivative
y := t → −cos(t);

# Define the error function as a function of h for a specific t
```

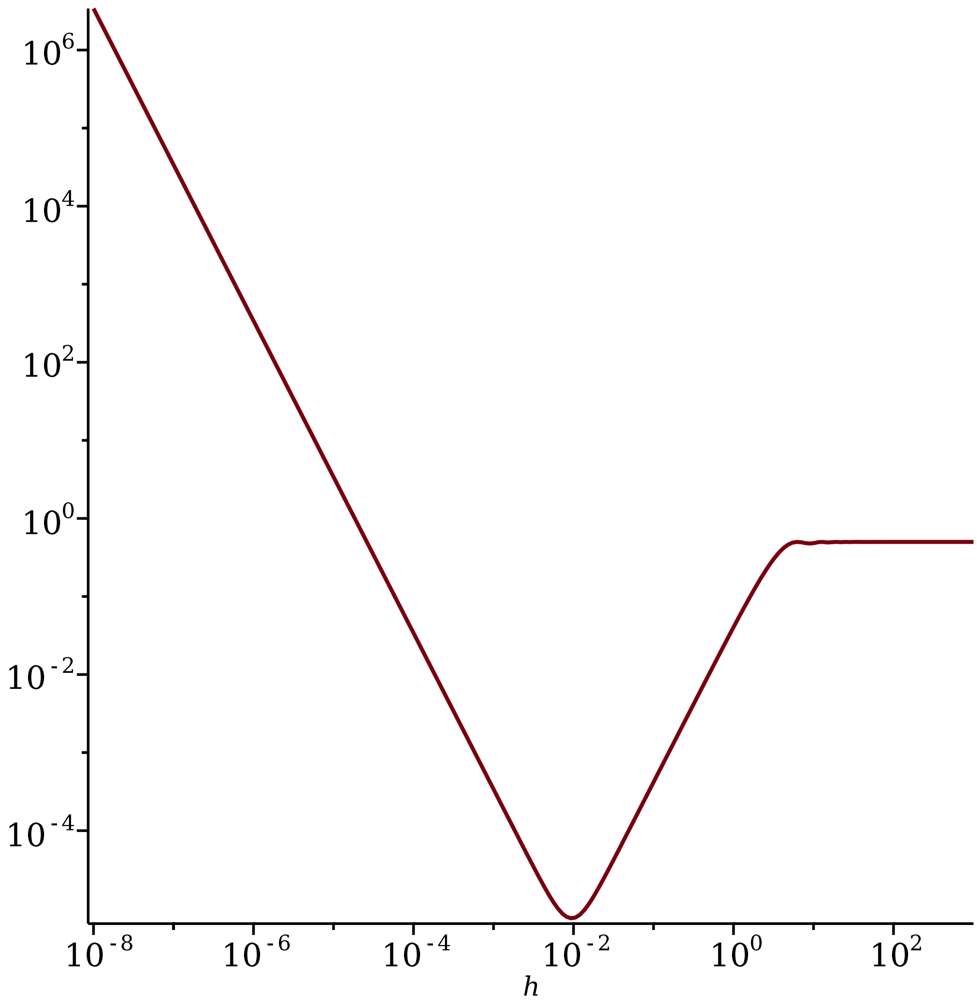
$$err := (h) \rightarrow \text{abs}\left(y(t\_val) - \frac{(\cos(h + t\_val) + \cos(h - t\_val) - 2*\cos(t\_val))}{h^2}\right);$$

```
# Plot the error as a function of h with log-log scale
loglogplot(err(h), h = 10^(−8) .. 10^3);
```

$$t\_val := 1.047197551$$

$$y := t \mapsto -\cos(t)$$

$$err := h \mapsto \left| y(t\_val) - \frac{\cos(h + t\_val) + \cos(h - t\_val) - 2\cdot\cos(t\_val)}{h^2} \right|$$


10^6
10^4
10^2
10^0
10^{-2}
10^{-4}
10^{-8}
10^{-6}
10^{-4}
10^{-2}
10^0
10^2
h

>