Om u är beräknat med tidssteget k, $\bar{u}$ är beräknat med tidssteget k/2.
Felluppskattning beräknas med:

$$|u - \bar{u}| \leq \frac{LT}{2} k$$

Generellt kan vi säga:

$$|u - \bar{u}| \leq C k^p$$

Vi tittar alltså på skillnaderna och försöker bestämma p:

$d_1 \approx 2{,}87 - 2{,}31 = 0{,}56$
$d_2 \approx 2{,}31 - 2{,}25 = 0{,}07$
$d_3 \approx 2{,}24 - 2{,}23 = 0{,}01$

Varje halvering av k ger c:a en faktor $\frac{1}{8} = \frac{1}{2^3}$.

Allstså kan vi uppskatta att metoden är av ordning 3.

$\vec{f}(\vec{x})=,$

där $\vec{\cdot}$ och $\vec{x}$ är vektorvärda (arrayer).

Newtons metod:

$J \cdot x_{i+1} = J \cdot x_i - f(x_i)$

$J = f'(x)$ (Jacobianen av f; Jabobimatris)

```
def newton(f, x0):
    def g(x):
        # Compute the Jacobian: f'(x)
        J = jacobian(f, x)
        # Compute right hand side of Newton
        # iteration and solve the linear system.
        r = dot(J, x) - f(x)
        return linear_solve(J, r)

    return fixedpoint(g, x0)

def fixedpoint(g, x0):
    TOL = 1.0e-10
    while (diff > TOL):
        y = g(x)
        diff = max_norm(y - x)
        x = y

def f(x):
    [...]

x0 = zeros(3)
x = newton(f, x0)
```

$$\mathbf{A}\vec{x}=\vec{b}$$

där

$$\mathbf{A}=\begin{pmatrix}10 & 1 & 2\\ 0 & 10 & 1\\ 1 & 2 & 20\end{pmatrix},\quad \vec{b}=\begin{pmatrix}10\\ 12\\ 24\end{pmatrix}$$

Vi definierar **D** som diagonalen av **A** och **M** = **A** − **D**.
I detta fall:

$$\mathbf{D}=\begin{pmatrix}10 & 0 & 0\\ 0 & 10 & 0\\ 0 & 0 & 20\end{pmatrix},\quad \mathbf{M}=\begin{pmatrix}0 & 1 & 2\\ 0 & 0 & 1\\ 1 & 2 & 0\end{pmatrix}$$

Jacobis metod kan då formuleras som:

$$x_1=\mathbf{D}^{-1}\left(-\mathbf{M}\vec{x}_0+\vec{b}\right)$$

där 1 och 0 är iterationsnummer.

Vi väljer $\vec{x}_0=\begin{pmatrix}1 & 2 & 3\end{pmatrix}^T=\begin{pmatrix}1\\ 2\\ 3\end{pmatrix}$,

vi har då att $\vec{x}_1=\mathbf{D}^{-1}\left(-\mathbf{M}\vec{x}_0+\vec{b}\right)=\begin{pmatrix}\frac{4}{10} & \frac{9}{10} & \frac{19}{20}\end{pmatrix}^T=\begin{pmatrix}4/10\\ 9/10\\ 19/20\end{pmatrix}$.

Vi vill lösa begynnelsevärdesproblemet: $\dot{u} = f(t; w)$

$$w = \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix}$$

$$\dot{w} = f(t;\, w) = \begin{pmatrix} f_0(t;\, w) \\ f_1(t;\, w) \end{pmatrix} = \begin{pmatrix} -{w_1}^2 w_0 + \sin(t) \\ -{w_0}^2 w_1 + \sin(2t) \end{pmatrix}$$

Formulera `timestep()` och `solve()` från modul 3 och anropa `solve()` med $f(t; w)$.

För att beräkna integralen $\int_0^b \left(u(t)^2 + v(t)^2\right) dt$ kan vi skriva:

$f_{energy}(t;\, z) = \left(u(t)^2 + v(t)^2\right)$ där vi stoppar in de beräknade lösningarna u(t) och v(t), genom exempelvis en funktion som `piecewise_linear_adapater()`. Sedan anropar vi `solve()` med $f_{energy}(t;\, z)$ för att beräkna integralen.