Numerical Solution of Differential Equations

B21/B1

Professor Endre Süli

Lecture 6

Initial value problem

$$y' = f(x,y)\,, \qquad x \in [x_0, X_M]$$

$$y(x_0) = y_0$$

Lecture 6

Initial value problem

$$y' = f(x,y)\,, \qquad x \in [x_0, X_M]$$

$$y(x_0) = y_0$$

General form of a linear multistep method:

Lecture 6

Initial value problem

$$y' = f(x,y)\,, \qquad x \in [x_0, X_M]$$

$$y(x_0) = y_0$$

General form of a linear multistep method:

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j}$$

$f_{n+j} = f(x_{n+j}, y_{n+j})$, $\alpha_k \neq 0$, $|\alpha_0| + |\beta_0| \neq 0$

Stiff problems: Van der Pol's oscillator

$$y'' + \mu(y^2 - 1)y' + y = 0\,, \qquad y(0) = 2, \quad y'(0) = 1$$

with $\mu = 0, 1, 2, 5, 10, 20, 50$. Solve on the interval $[0, 60]$.

Stiff problems: Van der Pol's oscillator

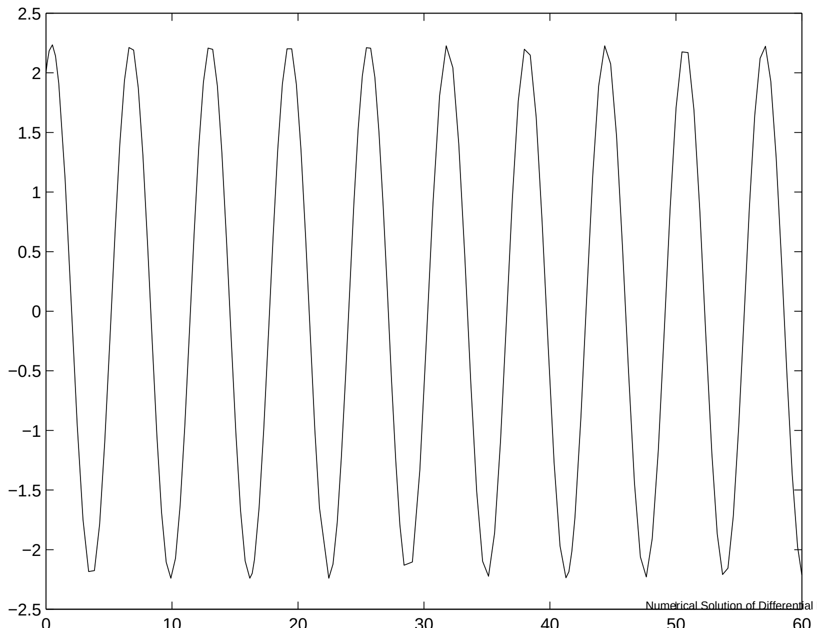
$$y'' + \mu(y^2 - 1)y' + y = 0\,, \qquad y(0) = 2, \quad y'(0) = 1$$

with $\mu = 0, 1, 2, 5, 10, 20, 50$. Solve on the interval $[0, 60]$.

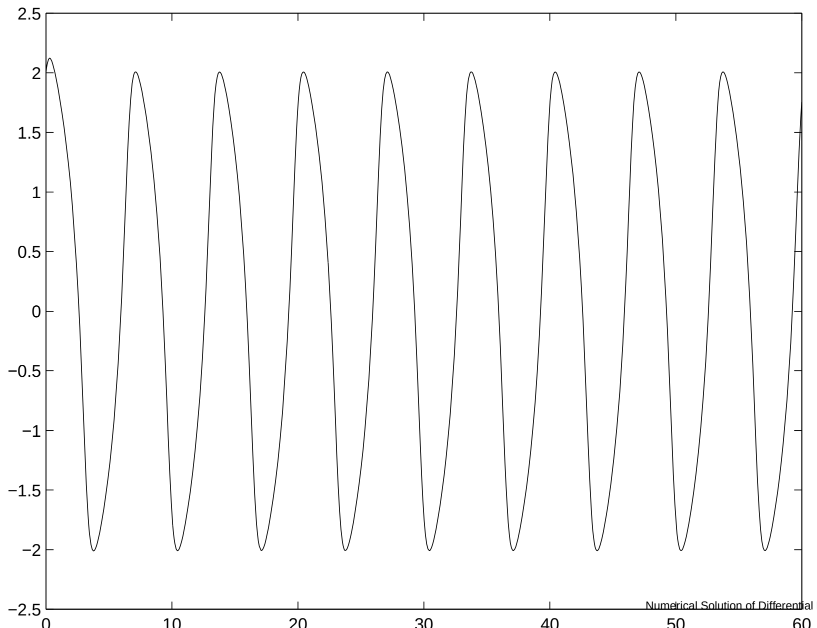
$$\begin{aligned} y_1' &= y_2\,, & y_1(0) &= 2 \\ y_2' &= \mu(1 - y_1^2)y_2 - y_1\,, & y_2(0) &= 1 \end{aligned}$$

Numerical Solution of Differential Equations – p.3/45

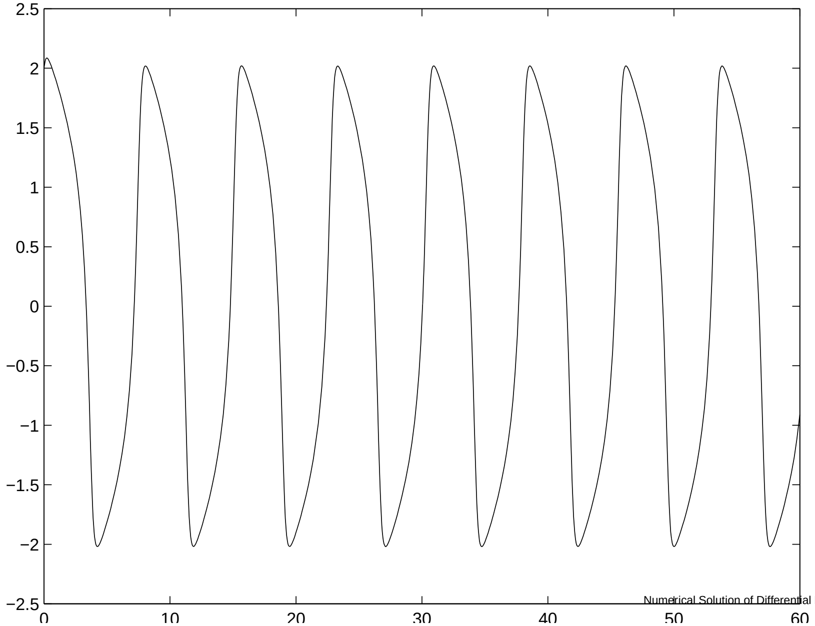
Result for $\mu = 0$: $\quad y = y_1$



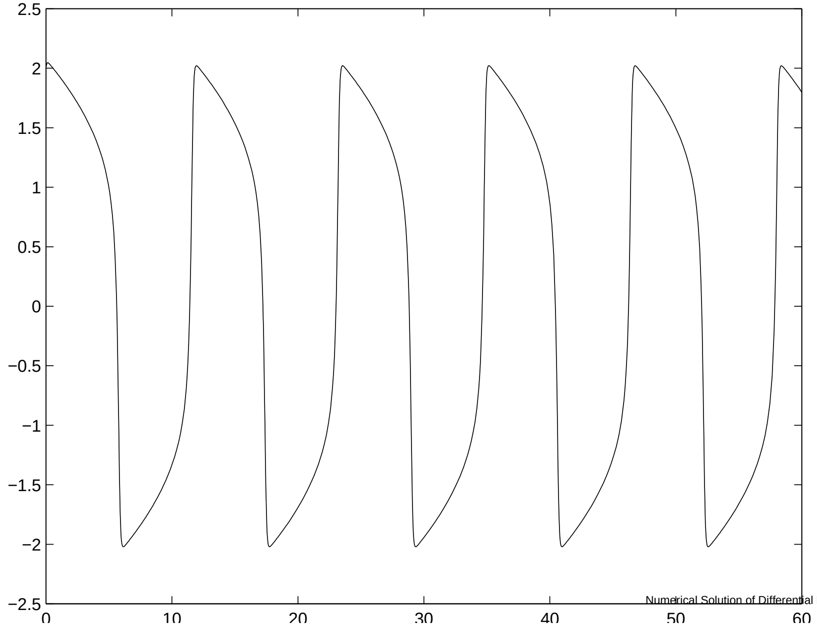
Result for $\mu = 1$: $\quad y = y_1$



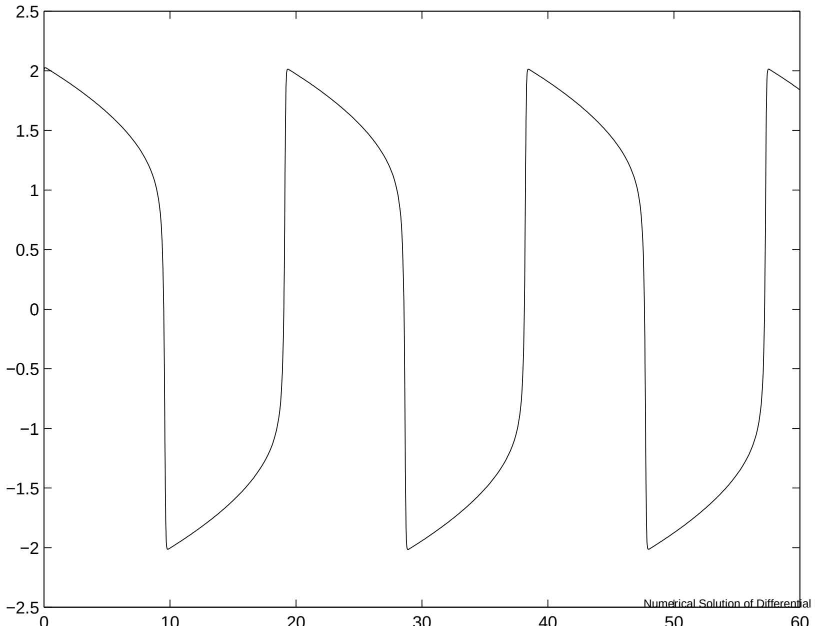
Result for $\mu = 2$: $y = y_1$



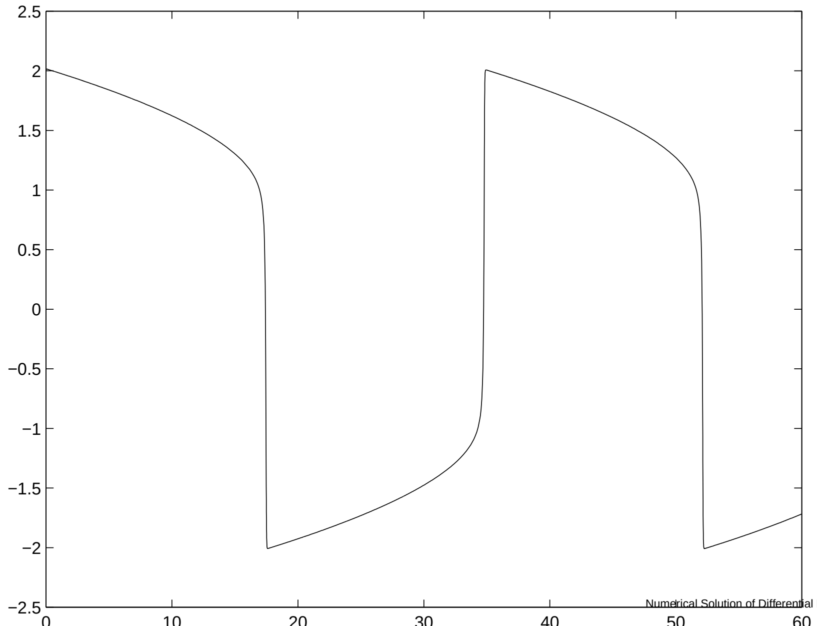
Result for $\mu = 5$: $y = y_1$



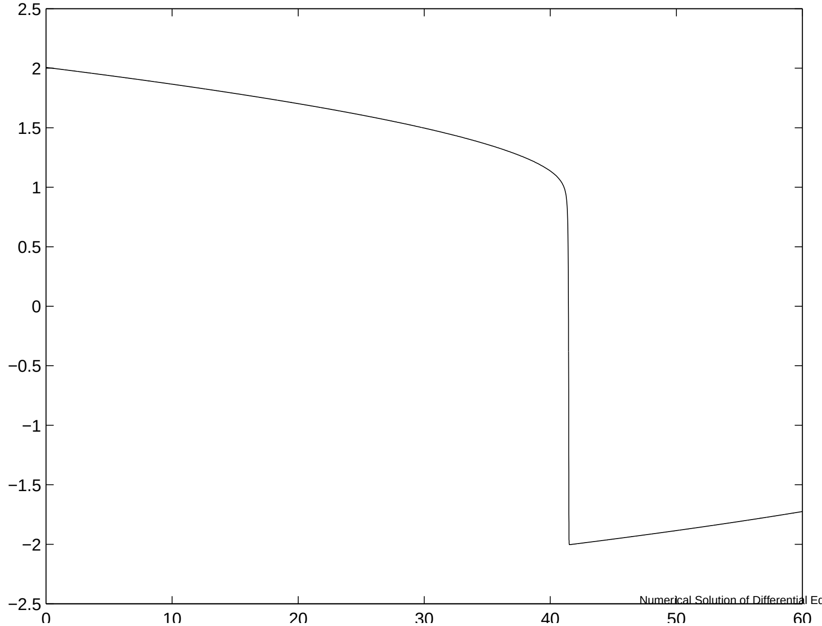
Result for $\mu = 10$: $y = y_1$



Result for $\mu = 20$: $y = y_1$



Result for $\mu = 50$: $\quad y = y_1$



A stiff linear system

Let $\lambda < 0$ and consider

$$y'' - (\lambda - 1)y' - \lambda y = 0\,, \qquad y(0) = 1, \quad y'(0) = -\lambda - 2$$

Write $y_1 = y\,, \quad y_2 = y'\,, \quad \mathbf{y} = (y_1, y_2)^{\mathrm{T}}$,

$$\mathbf{y}' = A\mathbf{y}\,, \qquad \mathbf{y}(0) = \mathbf{y}_0\,.$$

$$A = \begin{pmatrix} 0 & 1 \\ \lambda & \lambda - 1 \end{pmatrix} \quad \text{and} \quad \mathbf{y}_0 = \begin{pmatrix} 1 \\ -\lambda - 2 \end{pmatrix}.$$

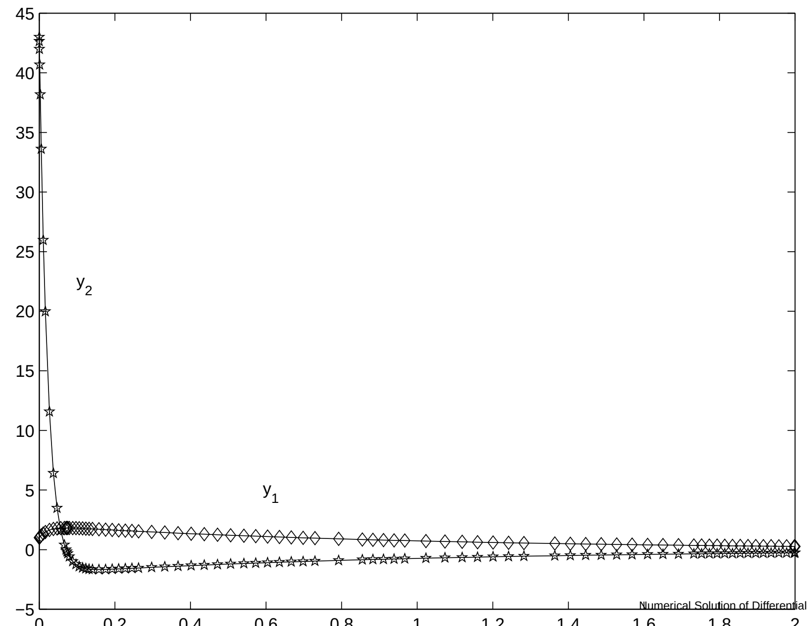
A stiff linear system

$$\det(A - zI) = z^2 - (\lambda - 1)z - \lambda$$

$$z_1 = -1\,, \qquad z_2 = \lambda$$

$$y_1(x) = 2\mathrm{e}^{-x} - \mathrm{e}^{\lambda x}\,, \qquad y_2(x) = -2\mathrm{e}^{-x} - \lambda \mathrm{e}^{\lambda x}$$

A stiff linear system: $\lambda = -45$



Matlab code (for $\lambda = -45$)

```
linear.m
function dy = lineq(x,y)
lambda=-45;
dy = zeros(2,1);
dy(1) = y(2);
dy(2) = lambda * y(1)+(lambda-1)*y(2);

launchlin.m
lambda=-45;
[x,y] = ode113('linear',[0 2],[1 -lambda-2]);
plot(x,y(:,1),'kd-', x, y(:,2),'kp-')
```


The simplest example

$$y' = \lambda y\,, \qquad y(0) = 1$$

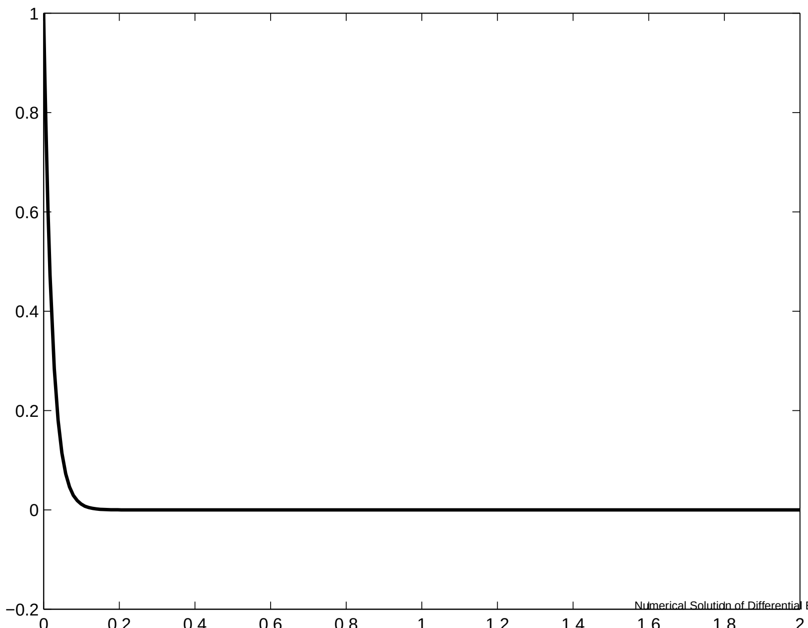
where

$$\mathrm{Re}(\lambda) < 0\,.$$

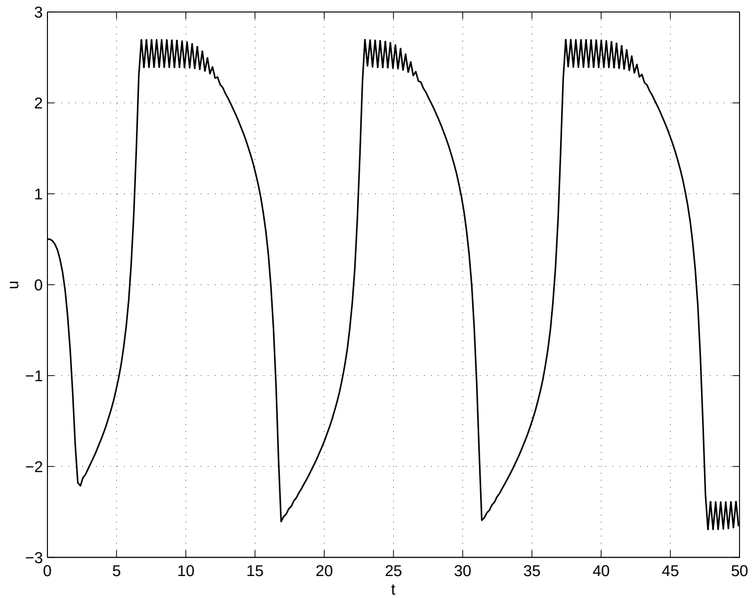
Exact solution

$$y(x) = \mathrm{e}^{\lambda x} = \mathrm{e}^{x\,\mathrm{Re}(\lambda)}\mathrm{e}^{\imath x\,\mathrm{Im}(\lambda)}$$

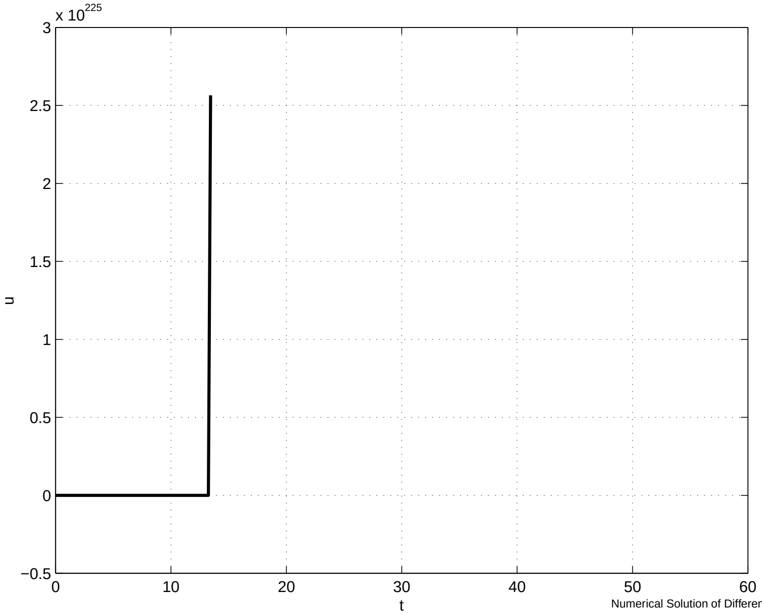
The simplest example: $y' = \lambda y$, $y(0) = 1$, for $\lambda = -45$



Van der Pol's equation by Euler's method: $h = 0.18353535$



Van der Pol's equation by Euler's method: $h = 0.184$



Matlab code for van der Pol's eq: $y'' + \mu(y^2 - 1)y' + y = 0$

```
h = input('h?  ');
mu = 2;
Xmax = 50;
nmax = round(Xmax/h);
y = [.5 0];
yvec = [y; zeros(nmax,2)];
xvec = zeros(nmax+1,1);
for i = 1:nmax
yprime = [y(2) -mu*(y(1).^2-1)*y(2)-y(1)];
y = y + h*yprime;
yvec(i+1,:)  = y;
xvec(i+1) = i*h;
end
plot(xvec,yvec(:,1),'linewidth',1),grid
```


What has gone wrong? ... Absolute stability

Now suppose we apply

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f(x_{n+j}, y_{n+j})$$

to this model problem ($f(x, y) = \lambda y$ with $\mathrm{Re}(\lambda) < 0$).

$$\sum_{j=0}^{k} (\alpha_j - \lambda h \beta_j) y_{n+j} = 0$$

Let us write $\bar{h} = \lambda h$. Clearly, $\mathrm{Re}(\bar{h}) < 0$.

Absolute stability

Then,

$$\sum_{j=0}^{k} (\alpha_j - \bar{h}\beta_j) y_{n+j} = 0$$

General solution

$$y_n = \sum_{s} p_s(n) z_s^n$$

where z_s is a zero of the stability polynomial

$$\pi(z, h) \equiv \rho(z) - \bar{h}\sigma(z) = \sum_{j=0}^{k} (\alpha_j - \bar{h}\beta_j) z^j$$

Absolute stability

Now,

$$\lim_{x\to\infty} y(x) = 0$$

Therefore, we want

$$\lim_{n\to\infty} y_n = 0$$

For this to hold, we need

$$|z_s| < 1 \qquad \text{for all } s = 1, \ldots, k.$$

Absolute stability

A linear multistep method is called *absolutely stable* in an open set $\mathcal{R}_A$ of the complex plane, if for all $\bar{h} \in \mathcal{R}_A$ all roots $z_s = z_s(\bar{h})$, $s = 1, \ldots, k$, of $\pi(z, \bar{h})$ satisfy $|z_s| < 1$.

The set $\mathcal{R}_A$ is called the *region of absolute stability*.

Exercise

Find the region of absolute stability of:

- Euler's method

Exercise

Find the region of absolute stability of:

- Euler's method
- The implicit Euler method

Region of absolute stability of Euler's method

$$y_{n+1} - y_n = hf(x_n, y_n)$$

Apply to $y' = \lambda y\,, \quad y(0) = 1$, where $\mathrm{Re}(\lambda) < 0$.

Region of absolute stability of Euler's method

$$y_{n+1} - y_n = hf(x_n, y_n)$$

Apply to $y' = \lambda y\,, \quad y(0) = 1$, where $\mathrm{Re}(\lambda) < 0$.

$$\rho(z) = z - 1\,, \qquad \sigma(z) = 1$$

Region of absolute stability of Euler's method

$$y_{n+1} - y_n = hf(x_n, y_n)$$

Apply to $y' = \lambda y\,, \quad y(0) = 1$, where $\mathrm{Re}(\lambda) < 0$.

$$\rho(z) = z - 1\,, \qquad \sigma(z) = 1$$

$$\pi(z, \bar{h}) = z - 1 - \bar{h}$$

Region of absolute stability of Euler's method

$$y_{n+1} - y_n = hf(x_n, y_n)$$

Apply to $y' = \lambda y\,, \quad y(0) = 1$, where $\mathrm{Re}(\lambda) < 0$.

$$\rho(z) = z - 1\,, \qquad \sigma(z) = 1$$

$$\pi(z, \bar{h}) = z - 1 - \bar{h} \quad \Rightarrow \quad z = 1 + \bar{h}$$

Region of absolute stability of Euler's method

$$y_{n+1} - y_n = hf(x_n, y_n)$$

Apply to $y' = \lambda y\,, \quad y(0) = 1$, where $\mathrm{Re}(\lambda) < 0$.

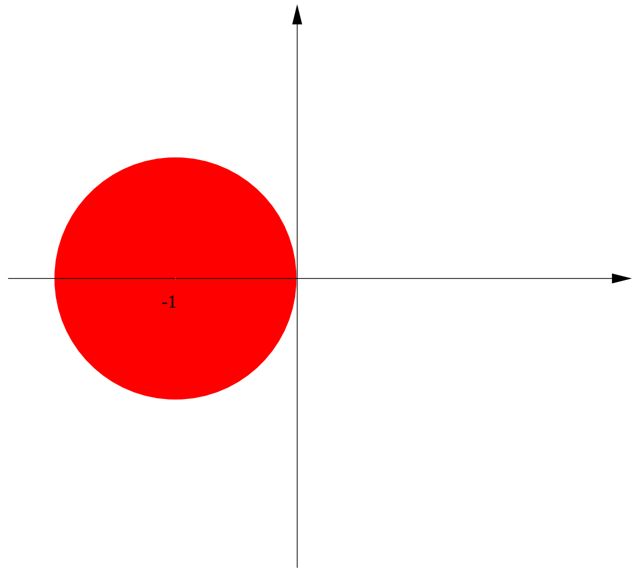
$$\rho(z) = z - 1\,, \qquad \sigma(z) = 1$$

$$\pi(z, \bar{h}) = z - 1 - \bar{h} \quad \Rightarrow \quad z = 1 + \bar{h}$$

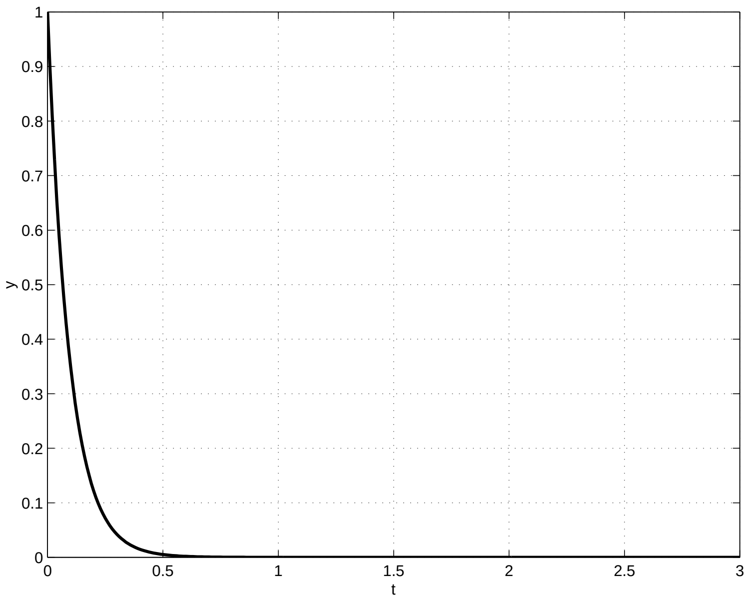
$$|z| < 1 \qquad \text{iff} \qquad |1 + \bar{h}| < 1$$

$$\mathcal{R}_A = \{\bar{h} \in \mathbb{C} \,:\, |1 + \bar{h}| < 1\}$$

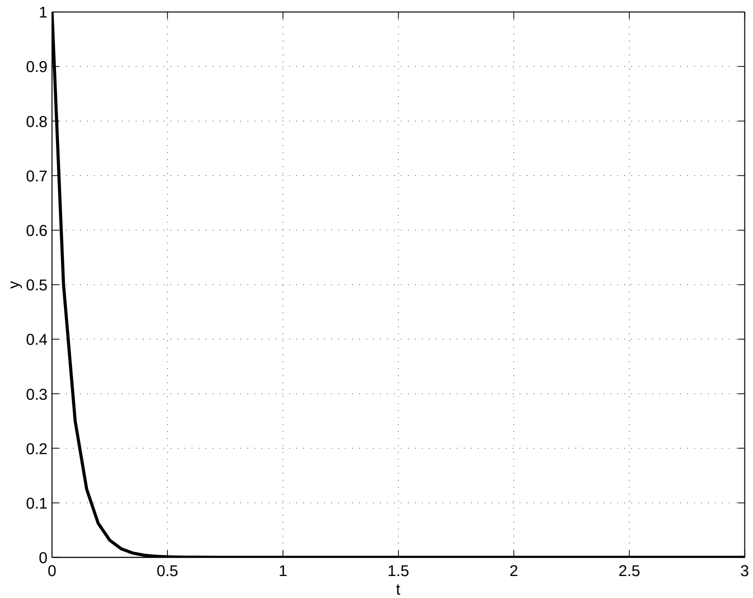
Region of absolute stability: interior of circle



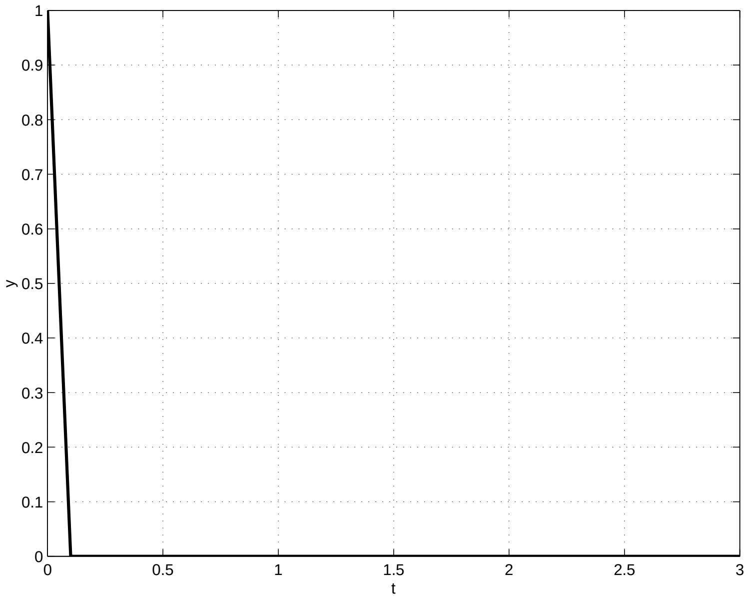
$y' = \lambda y\,, \quad y(0) = 1, \quad \lambda = -10, \qquad h = 0.01$



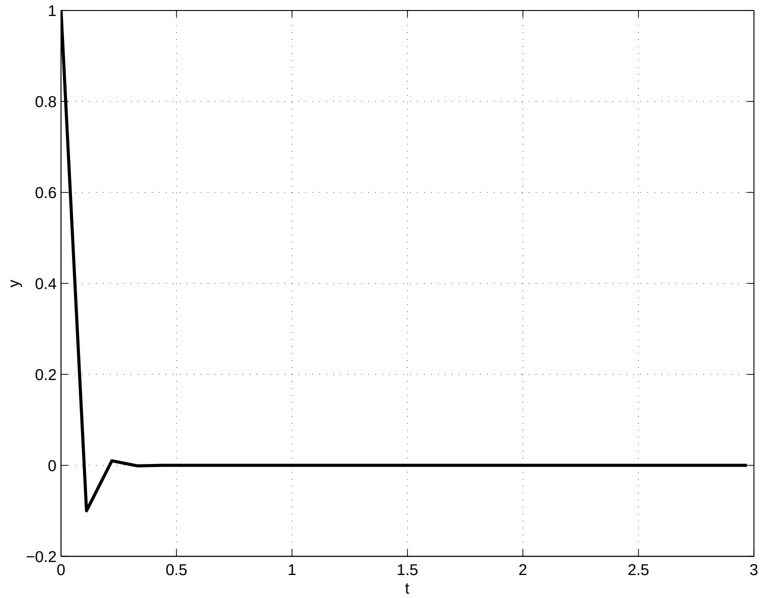
$y' = \lambda y\,, \quad y(0) = 1, \quad \lambda = -10, \qquad h = 0.05$



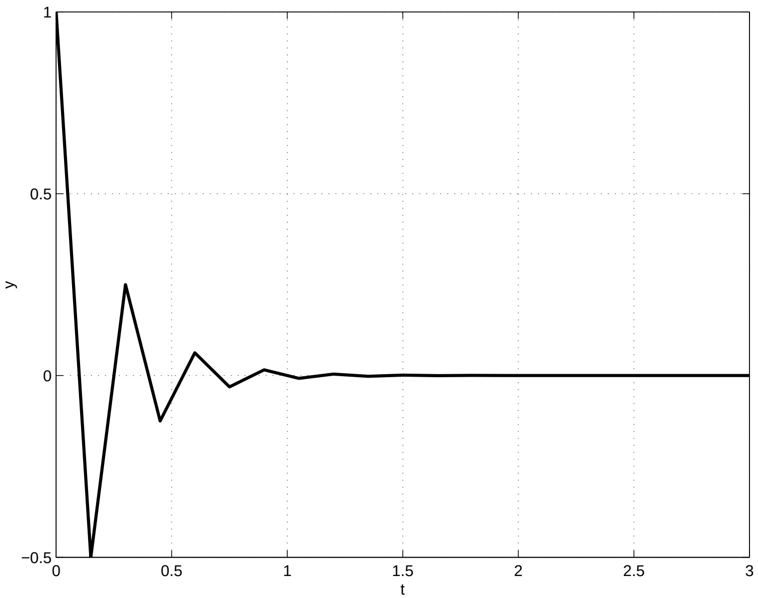
$y' = \lambda y\,, \quad y(0) = 1, \quad \lambda = -10, \qquad h = 0.1$



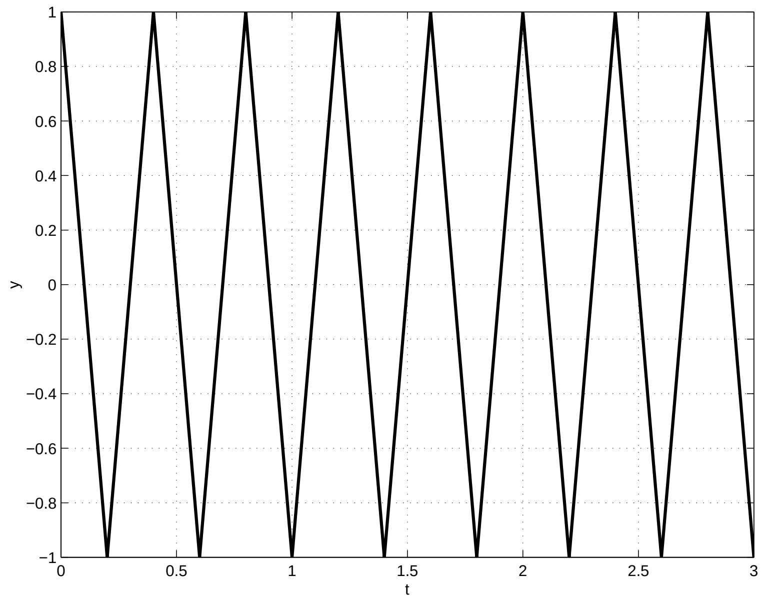
$y' = \lambda y\,, \quad y(0) = 1, \quad \lambda = -10, \qquad h = 0.11$



$y' = \lambda y\,, \quad y(0) = 1, \quad \lambda = -10, \qquad h = 0.15$



$y' = \lambda y\,, \quad y(0) = 1, \quad \lambda = -10, \qquad h = 0.2$



Matlab code for $y' = \lambda y$, $y(0) = 1$, by Euler's method

```
h=input('h?  ');
lambda = -10;
Xmax = 3;
nmax = round(Xmax/h);
y = 1;
yvec = [y; zeros(nmax,1)];
xvec = zeros(nmax+1,1);
for i = 1:nmax
yprime = lambda*y(1);
y = y + h*yprime;
yvec(i+1,:)  = y;
xvec(i+1) = i*h;
end
plot(xvec,yvec(:,1),'k-','linewidth',2), grid
```


Absolute stability of the implicit Euler method

$$y_{n+1} - y_n = hf(x_{n+1}, y_{n+1})$$

Apply to $y' = \lambda y\,, \quad y(0) = 1$, where $\mathrm{Re}(\lambda) < 0$.

Absolute stability of the implicit Euler method

$$y_{n+1} - y_n = hf(x_{n+1}, y_{n+1})$$

Apply to $y' = \lambda y\,, \quad y(0) = 1$, where $\mathrm{Re}(\lambda) < 0$.

$$\rho(z) = z - 1\,, \qquad \sigma(z) = z$$

Absolute stability of the implicit Euler method

$$y_{n+1} - y_n = hf(x_{n+1}, y_{n+1})$$

Apply to $y' = \lambda y\,, \quad y(0) = 1$, where $\mathrm{Re}(\lambda) < 0$.

$$\rho(z) = z - 1\,, \qquad \sigma(z) = z$$

$$\pi(z, \bar{h}) = z - 1 - \bar{h}z$$

Absolute stability of the implicit Euler method

$$y_{n+1} - y_n = hf(x_{n+1}, y_{n+1})$$

Apply to $y' = \lambda y\,, \quad y(0) = 1$, where $\mathrm{Re}(\lambda) < 0$.

$$\rho(z) = z - 1\,, \qquad \sigma(z) = z$$

$$\pi(z, \bar{h}) = z - 1 - \bar{h}z \quad \Rightarrow \quad z = \frac{1}{1 - \bar{h}}$$

Absolute stability of the implicit Euler method

$$y_{n+1} - y_n = hf(x_{n+1}, y_{n+1})$$

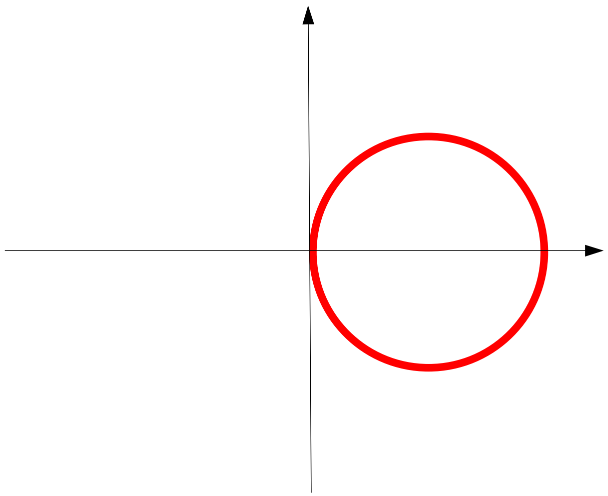
Apply to $y' = \lambda y\,, \quad y(0) = 1$, where $\mathrm{Re}(\lambda) < 0$.

$$\rho(z) = z - 1\,, \qquad \sigma(z) = z$$

$$\pi(z, \bar{h}) = z - 1 - \bar{h}z \quad \Rightarrow \quad z = \frac{1}{1 - \bar{h}}$$

$$\mathcal{R}_A = \{\bar{h} \in \mathbb{C} \,:\, |1 - \bar{h}| > 1\}$$

Region of absolute stability: exterior of circle



A–stability

A linear multistep method is said to be A–stable if its region of absolute stability, $\mathcal{R}_A$, contains the whole of the open left-hand complex half–plane, $\mathrm{Re}(\bar{h}) < 0$.

Example: The implicit Euler method is A–stable.

The Dahlquist Barrier Theorem

- No explicit linear multistep method is A–stable.

The Dahlquist Barrier Theorem

- No explicit linear multistep method is A–stable.
- The order of an A–stable implicit linear multistep method is ≤ 2.

The Dahlquist Barrier Theorem

- No explicit linear multistep method is A–stable.
- The order of an A–stable implicit linear multistep method is ≤ 2.
- The second–order A–stable linear multistep method with smallest error constant is the trapezium rule method.

Stepsize control: The Milne device

Suppose we are using the pth order linear k-step method

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f(x_{n+j}, y_{n+j})$$

How to choose h appropriately?

Stepsize control: The Milne device

Suppose we are using the pth order linear k-step method

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f(x_{n+j}, y_{n+j})$$

How to choose h appropriately?

Consider another pth order method:

$$\sum_{j=0}^{k} \tilde{\alpha}_j \tilde{y}_{n+j} = h \sum_{j=0}^{k} \tilde{\beta}_j f(x_{n+j}, \tilde{y}_{n+j})$$

The Milne device

If $y_n, y_{n+1}, \ldots, y_{n+k-1}$ are error free, then

$$y(x_{n+k}) - y_{n+k} = ch^{p+1} y^{(p+1)}(x_{n+k}) + \mathcal{O}(h^{p+2}) \qquad h \to 0$$

c is called the *local error constant*. [Proof at the back.]

o

The Milne device

If $y_n, y_{n+1}, \ldots, y_{n+k-1}$ are error free, then

$$y(x_{n+k}) - y_{n+k} = ch^{p+1} y^{(p+1)}(x_{n+k}) + \mathcal{O}(h^{p+2}) \qquad h \to 0$$

c is called the *local error constant*. [Proof at the back.]

o

Similarly,

$$y(x_{n+k}) - \tilde{y}_{n+k} = \tilde{c}h^{p+1} y^{(p+1)}(x_{n+k}) + \mathcal{O}(h^{p+2}) \qquad h \to 0$$

Suppose $\tilde{c} \neq c$.

The Milne device

Then,

$$\tilde{y}_{n+k} - y_{n+k} \approx (c - \tilde{c}) h^{p+1} y^{(p+1)}(x_{n+k})$$

Hence,

$$y(x_{n+k}) - y_{n+k} \approx \frac{c}{c - \tilde{c}} (\tilde{y}_{n+k} - y_{n+k})$$

The Milne device

Then,

$$\tilde{y}_{n+k} - y_{n+k} \approx (c - \tilde{c}) h^{p+1} y^{(p+1)}(x_{n+k})$$

Hence,

$$y(x_{n+k}) - y_{n+k} \approx \frac{c}{c - \tilde{c}} (\tilde{y}_{n+k} - y_{n+k})$$

Define local refinement indicator:

$$\kappa = \left| \frac{c}{c - \tilde{c}} \right| \, |\tilde{y}_{n+k} - y_{n+k}|$$

The Milne device: local refinement criteria

- Error control per step
  Given a fixed tolerance δ, require that

$$\kappa \leq \delta$$

The Milne device: local refinement criteria

- Error control per step
  Given a fixed tolerance δ, require that

$$\kappa \leq \delta$$

- Error control unit per step
  Given a fixed tolerance δ, require that

$$\kappa \leq h\delta$$

Numerical Solution of Differential Equations – p.41/45

The Milne device: the adaptive algorithm

1. Set h

The Milne device: the adaptive algorithm

1. Set h
2. Compute new y

The Milne device: the adaptive algorithm

1. Set h
2. Compute new y
3. Is $\kappa \leq h\delta$?

The Milne device: the adaptive algorithm

1. Set h
2. Compute new y
3. Is $\kappa \leq h\delta$?

- NO: halve h, remesh and GOTO 2

The Milne device: the adaptive algorithm

1. Set h
2. Compute new y
3. Is $\kappa \leq h\delta$?

- NO: halve h, remesh and GOTO 2
- YES: GOTO 4

The Milne device: the adaptive algorithm

1. Set h
2. Compute new y
3. Is $\kappa \le h\delta$?
   - NO: halve h, remesh and GOTO 2
   - YES: GOTO 4
4. Is $x \ge X_M$?

The Milne device: the adaptive algorithm

1. Set h
2. Compute new y
3. Is $\kappa \le h\delta$?
   - NO: halve h, remesh and GOTO 2
   - YES: GOTO 4
4. Is $x \ge X_M$?
   - YES: STOP

The Milne device: the adaptive algorithm

1. Set h
2. Compute new y
3. Is $\kappa \le h\delta$?
   - NO: halve h, remesh and GOTO 2
   - YES: GOTO 4
4. Is $x \ge X_M$?
   - YES: STOP
   - NO: GOTO 5

The Milne device: the adaptive algorithm

1. Set h
2. Compute new y
3. Is $\kappa \le h\delta$?
   - NO: halve h, remesh and GOTO 2
   - YES: GOTO 4
4. Is $x \ge X_M$?
   - YES: STOP
   - NO: GOTO 5
5. Is $\kappa \le \frac{1}{10}h\delta$?

The Milne device: the adaptive algorithm

1. Set h
2. Compute new y
3. Is $\kappa \le h\delta$?
- NO: halve h, remesh and GOTO 2
- YES: GOTO 4
4. Is $x \ge X_M$?
- YES: STOP
- NO: GOTO 5
5. Is $\kappa \le \frac{1}{10}h\delta$?
- YES: double h, advance x, remesh, GOTO 2

The Milne device: the adaptive algorithm

1. Set h
2. Compute new y
3. Is $\kappa \le h\delta$?
- NO: halve h, remesh and GOTO 2
- YES: GOTO 4
4. Is $x \ge X_M$?
- YES: STOP
- NO: GOTO 5
5. Is $\kappa \le \frac{1}{10}h\delta$?
- YES: double h, advance x, remesh, GOTO 2
- NO: advance x, GOTO 2

Proof

$$\alpha_k y_{n+k} - h\beta_k f(x_{n+k}, y_{n+k}) + \sum_{j=0}^{k-1} \alpha_j y_{n+j} - h \sum_{j=0}^{k-1} \beta_j f(x_{n+j}, y_{n+j}) = 0$$

By the definition of the truncation error T_n, and writing $B = \sum_{j=0}^{k} \beta_j$,

$$\alpha_k y(x_{n+k}) - h\beta_k f(x_{n+k}, y(x_{n+k})) + \sum_{j=0}^{k-1} \alpha_j y(x_{n+j}) - h \sum_{j=0}^{k-1} \beta_j f(x_{n+j}, y(x_{n+j})) = B\,h\,T_n$$

Proof

Since $y_n, y_{n+1}, \ldots, y_{n+k-1}$ are error free,

$$\alpha_k \left[y(x_{n+k}) - y_{n+k}\right] - h\beta_k \left[f(x_{n+k}, y(x_{n+k})) - f(x_{n+k}, y_{n+k})\right] = B\,h\,T_n$$

$$\alpha_k \left[y(x_{n+k}) - y_{n+k}\right] - h\beta_k \frac{\partial f}{\partial y}(x_{n+k}, \eta_{n+k}) \left[y(x_{n+k}) - y_{n+k}\right] = B\,h\,T_n$$

$$\left[\alpha_k - h\beta_k \frac{\partial f}{\partial y}(x_{n+k}, \eta_{n+k})\right] \left(y(x_{n+k}) - y_{n+k}\right) = B\,h\,T_n$$

Proof

As $h \to 0$,

$$\alpha_k - h\beta_k \frac{\partial f}{\partial y}(x_{n+k}, \eta_{n+k}) \to \alpha_k \neq 0$$

Therefore,

$$y(x_{n+k}) - y_{n+k} = \text{Const.}\, h\, T_n$$

Recalling that

$$T_n = \text{Const.} h^p y^{(p+1)}(x_{n+k}) + \mathcal{O}(h^{p+1})$$

we deduce the desired result. o