Prof. D. Kressner
M. Steinlechner

1 ► First Runge-Kutta methods

$$y'(t) = \lambda y(t), \quad y(0) = y_0 \qquad (*)$$

a) Runge's method: $y_1 = y_0 + h f\left(t_0 + \frac{h}{2}, y_0 + \frac{h}{2} f(t_0, y_0)\right)$

$$= y_0 + h\lambda\left(y_0 + \frac{h}{2}\lambda y_0\right)$$

$$= \left(1 + h\lambda + \frac{(h\lambda)^2}{2}\right) y_0$$

b) In general, it is easy to see that an explicit s-stage Runge-Kutta method applied to $(*)$ will produce a polynomial y_1 in h of degree at most s after one step of length h.

Since the order is s, the Taylor polynoms of order s around t_0 of $h \mapsto y(t_0+h)$ and $h \mapsto y_1(h)$ [here $t_0 = 0$] have to agree in

Since $y(h) = e^{\lambda h}$, it follows

$$y_1(h) = \sum_{k=0}^{s} \frac{\lambda^k}{k!} h^k.$$

$\square$

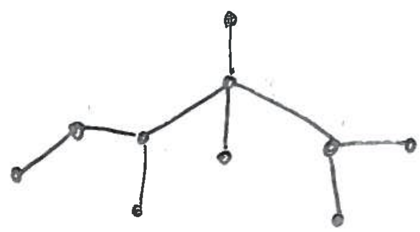
2 ► Rooted trees

i) $[\odot, [\odot], [\odot, \odot]]$ correspond to the tree

It represent the differential $f'''(f, f'f, f''(f,f))$

$\beta ! = 7 \cdot 2 \cdot 3 = 42$

ii) $[[[[\odot], \odot], \odot, [\odot, \odot]]]$



$f' f'''(f''(f'f, f), f, f''(f, f))$

$\beta ! = 10 \cdot 9 \, (4 \cdot 2) \cdot 3 = 2160$

3 ► Verifying order of Runge-Kutta methods

Problem 1

a)

$$\begin{array}{c|ccc} c_1 & 0 & & \\ c_2 & a_{21} & 0 & \\ c_3 & a_{31} & a_{32} & 0 \\ \hline & b_1 & b_2 & b_3 \end{array}$$

Consistency conditions according to Table 2.1 (cf. Example 2.13):

$$\sum_i b_i = 1 \,, \quad \sum_i b_i c_i = \frac{1}{2} \,, \quad \sum_i b_i c_i^2 = \frac{1}{3} \,, \quad \sum_{i,j} b_i a_{ij} c_j = \frac{1}{6}$$

Autonomization invariance: $\sum_j a_{ij} = c_i$ for every i

Explicitly:

$$c_1 = 0 \qquad\qquad b_1 + b_2 + b_3 = 1$$

$$c_2 = a_{21} \qquad\qquad b_2 c_2 + b_3 c_3 = \frac{1}{2}$$

$$c_3 = a_{31} + a_{32} \qquad\qquad b_2 c_2^2 + b_3 c_3^2 = \frac{1}{3}$$

$$b_3 a_{32} c_2 = \frac{1}{6}$$

b)

$$\begin{array}{c|ccc} 0 & 0 & & \\ \frac{1}{2} & \frac{1}{2} & 0 & \\ 1 & -\frac{1}{2} & \frac{3}{2} & 0 \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

Only the last of the above equations is violated. It should hold $a_{32} = \frac{1}{6 c_2 b_3} = 2$, but then $a_{31} = c_3 - a_{32} = -1$. The correct table is

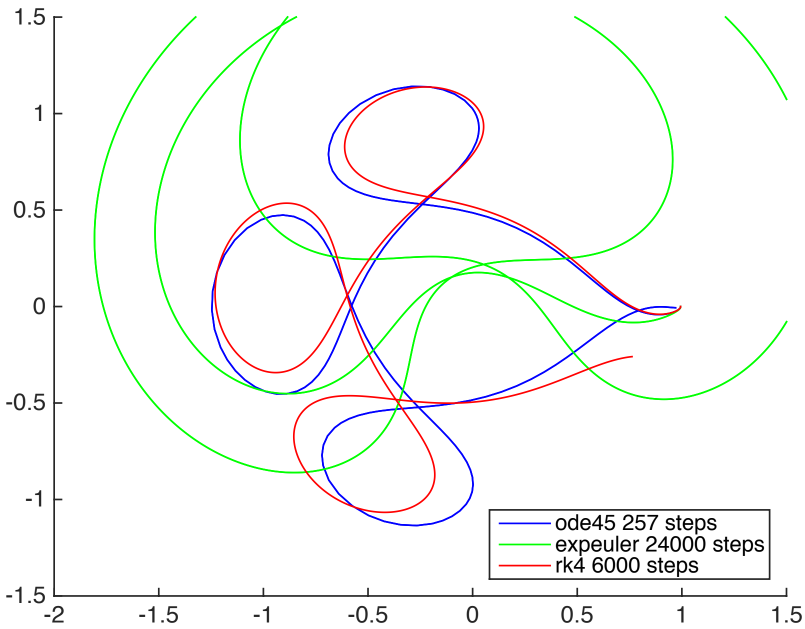
$$\begin{array}{c|ccc} 0 & 0 & & \\ \frac{1}{2} & \frac{1}{2} & 0 & \\ 1 & -1 & 2 & 0 \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

4 ▶ A problem from astronomy

a) Runge-Kutta RK4:

```
function [t,y] = rk4(f,tspan,y0,N)

h = (tspan(2)-tspan(1))/N;
t = tspan(1)*ones(1,N+1) + h*(0:N);

t(1) = tspan(1);
y(:,1) = y0;

g=[1/6 1/3 1/3 1/6];

for i=2:N+1
    k1 = f( t(i-1), y(:,i-1) );
    k2 = f( t(i-1)+h/2, y(:,i-1)+(h/2)*k1 );
    k3 = f( t(i-1)+h/2, y(:,i-1)+(h/2)*k2 );
    k4 = f( t(i-1)+h, y(:,i-1)+h*k3 );

    y(:,i) = y(:,i-1) + h*(g(1)*k1+g(2)*k2+g(3)*k3+g(4)*k4);
end

end
```

b) Solving the astronomy problem:




```
function ex3problem3

    % specifying paramters
    mu =  0.012277471;
    y0 = [0.994; 0; 0; -2.00158510637908252240537862224];
    tspan = [0, 17.0652165601579625588917206249];
    fun = @(t,z) rhs(t,z,mu);

    %run expeuler
    Nexpeuler = 24000;
    [t1,y1] = expeuler(fun,tspan,y0,Nexpeuler);

    % run rk4
    Nrk4 = 6000;
    [t2,y2] = rk4(fun,tspan,y0,Nrk4);
```

```

    % Part (c)
    % --------

    %run ode45
    [t3,y3] = ode45(fun,tspan,y0);
    nos = length(t3);

    % all together
    figure(1); hold on
    axis([-2 1.5 -1.5 1.5])

    plot(y3(:,1),y3(:,2),'b');
    plot(y1(1,:),y1(2,:),'g');
    plot(y2(1,1),y(2,1),'r'); %dummy to get the legend

    legend(sprintf('ode45 %i steps',nos),...
        sprintf('expeuler %i steps',Nexpeuler),...
        sprintf('rk4 %i steps',Nrk4),...
        'location','southeast');

    %figure(1); comet(y2(1,:),y2(2,:)); hold on;

    % now plot all components of "exact" ode45 solution
    figure(2)
    plot(t3,y3(:,1),t3,y3(:,2),t3,y3(:,3),t3,y3(:,4));
    legend('x1','x2','v1','v2','location','southeast')

end

%Right hand side of the problem
function dotz = rhs(t,z,mu)
    dotz = zeros(4,1);
    y1 = z(1); y2 = z(2); v1 = z(3); v2 = z(4);
    D1 = (1 - mu)/( (y1 + mu)^2 + y2^2 )^(3/2);
    D2 = mu/( (y1 - (1 - mu))^2 + y2^2 )^(3/2);

    dotz(1) = v1; dotz(2) = v2; %due to order reduction

    dotz(3) = y1 + 2*v2 - D1*(y1 + mu) - D2*(y1 - (1 - mu));
    dotz(4) = y2 - 2*v1 - D1*y2 - D2*y2;
end

function [t,y] = expeuler(f,tspan,y0,N)
    y(:,1) = y0;
    h = (tspan(2)-tspan(1))/N;
    t = tspan(1)*ones(1,N+1) + h*(0:N);
    for i =2:N+1
        y(:,i) = y(:,i-1) + h*f(t(i-1),y(:,i-1));
    end
end
```