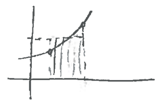
1 ► Order of implicit methods

For autonomous IVPs, find and prove the consistency order of the implicit midpoint rule given by the table $\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array}$.

Implicit midpoint rule for autonomous IVP:

$$k_1 = f(y_0 + \tfrac{h}{2} k_1)$$

$$y_1 = y_0 + h k_1$$

(so basically: $y_1 = y_0 + f(\tfrac{1}{2}(y_0 + y_1))$ )

Assume that $f \in C^1$ and k_1 is a unique continuous function of h (true by Theorem 3.8). Then k_1 is locally bounded and actually C^1 since it holds

$$\begin{aligned} k_1 &= f(y_0 + \tfrac{h}{2} f(y_0 + \tfrac{h}{2} k_1)) \\ &= f(y_0 + \tfrac{h}{2} f(y_0) + \mathcal{O}(\|h\|^2)) \\ &= f(y_0) + \frac{\partial f(y_0)}{\partial y}[\tfrac{h}{2} f(y_0)] + \mathcal{O}(\|h\|^2). \end{aligned}$$

This gives

$$y_1 = y_0 + h f(y_0) + \frac{h^2}{2} \frac{\partial f(y_0)}{\partial y}[f(y_0)] + \mathcal{O}(\|h\|^3)$$

The first three terms equal the corresponding ones in a Taylor expansion of the exact solution $y(t_0+h)$.

Hence: $y_1 - y(t_0+h) = \mathcal{O}(\|h\|^3)$

→ The midpoint scheme is of order $p = 2$.

(Find an example that shows that the order is not higher!)

2 ► Implementation of implicit RK methods

```
% Exercise 5 Problem 2
% ====================

function ex5problem2
close all;

% Define example problem

f = @(t,y) [-0.04*y(1) + 1e04*y(2)*y(3);...
    0.04*y(1) - 1e04*y(2)*y(3) - 3e07 * y(2)^2;...
    3e07*y(2)^2];
```

```
fjac = @(t,y) [-0.04, 1e04*y(3), 1e04*y(2);...
    0.04, -1e04*y(3) - 6e07*y(2), -1e04*y(2);...
    0, 6e07*y(2), 0];

sol = [0.988673939; 0.0000344771574; 0.0112915834]; % for T = 0.3

tspan = [0 0.3];
y0 = [1 0 0]';

N = [20,50,100,200];

% Test series
%
% [t1,y1] = itrapez(f,tspan,y0,20,fjac,1e-4,10);
% plot(t1,y1(2,:));
% title(sprintf('itrapez N=%i, abstol = 1e-4',N(i)));
%
% return


figure(1);
for i=1:4

% Run implicit trapezoid rule on example and plot second component
[t1,y1] = itrapez(f,tspan,y0,N(i),fjac,1e-4,10);
subplot(3,4,i)
plot(t1,y1(2,:));
title(sprintf('itrapez N=%i, abstol = 1e-4',N(i)));

% Run implicit trapezoid rule on example
[t2,y2] = itrapez(f,tspan,y0,N(i),fjac,1e-6,10);
subplot(3,4,i+4)
plot(t2,y2(2,:));
title(sprintf('itrapez N=%i, abstol = 1e-6',N(i)));


% Run classical RK4
[t3,y3] = rk4(f,tspan,y0,N(i));
subplot(3,4,i+8)
plot(t3,y3(2,:));
title(sprintf('RK4 N=%i',N(i)));
end

% ode45
disp('ODE45'); figure(2);
options = odeset('Stats','on','OutputSel',[2]);
ode45(f,[0,0.3],[1 0 0]',options);
disp(' ');
title('ODE45')

% ode23s without Jacobian information
disp('ODE23s'); figure(3);
ode23s(f,[0,0.3],[1 0 0]',options); figure;
disp(' ');
title('ODE23s')

% ode23s with Jacobian
disp('ODE23s with Jacobian'); figure(4);
options = odeset(options,'Jacobian',fjac);
ode23s(f,[0,0.3],[1 0 0]',options);
title('ODE23s Jac')

end

% Required functions
% ------------------

% Implict trapezoidal rule

```

```
function [t,y] = itrapez(f,tspan,y0,N,fjac,atol,maxiter)

h = (tspan(2)-tspan(1))/N;
t = tspan(1)*ones(1,N+1) + h*(0:N);
d = length(y0);

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

for i=2:N+1

    yn = y(:,i-1); tn = t(i-1);
    F = @(Z) Z - h*[zeros(d,1); f(tn,yn + Z(1:d))/2 + f(tn + h,yn + Z(d+1:2*d))/2];
    dF = @(Z) eye(2*d) - h*[zeros(d,2*d);...
        fjac(tn,yn + Z(1:d))/2, fjac(tn + h,yn + Z(d+1:2*d))/2];

    Z = newton(F,dF,zeros(2*d,1),atol,maxiter);

    y(:,i) = yn + h*( f(tn,yn + Z(1:d))/2 + f(tn + h,yn + Z(d+1:2*d))/2 );
end

end

% Newton method

function Z = newton(F,dF,Z0,atol,maxiter)

Z = Z0;
theta = 0.5;
i=1;

Delta = - dF(Z) \ F(Z);
Z = Z + Delta;

while norm(Delta)*theta/(1-theta) > atol && i < maxiter

    Delta_old = Delta;
    Delta = - dF(Z) \ F(Z);
    Z = Z + Delta;
    theta = norm(Delta)/norm(Delta_old);
    i = i+1;

end

end

% Classical 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
```