Yannis Bähni

Numerical Analysis

In Applications and Examples

Yannis Bähni

Numerical Analysis

In Applications and Examples

CHAPTER 1

Ordinary Differential Equations

1. Extrapolation and Step Size Control

1.1. The Implicit Trapezoidal Rule. The *implicit trapezoidal rule* is given by

$$y_1 = y_0 + \frac{h}{2}\left(f(x_0, y_0) + f(x_1, y_1)\right) \tag{1}$$

As one observes, the discretization y_1 of $y(x_1)$ is calculated implicitely. There are two simple approaches to handle this impliciteness. First of all, y_1 is a fixed point of the function

$$\Phi(y; x_0, y_0, h) := y_0 + \frac{h}{2}\left(f(x_0, y_0) + f(x_0 + h, y)\right) \tag{2}$$

Hence we may apply a simple *fixed point iteration* of the form

$$y_1^{(k+1)} = \Phi\left(y_1^{(k)}; x_0, y_0, h\right) \qquad k = 0, 1, 2, \ldots \tag{3}$$

until a certain tolerance or a user-specified maximal number of iterations is reached. An implementation of the fixed point iteration and the implicit trapezoidal rule can be found in listing 1 and 2 respectively.

```
function [ x ] = fixediter( phi,x0,tol,maxit )
x1 = phi(x0);
it = 1;
while norm(x1 - x0) > tol && it < maxit
    x0 = x1;
    x1 = phi(x0);
    it = it + 1;
end
x = x0;
end
```

Listing 1. Fixed point iteration.

The second approach would be applying the *Newton iteration*

$$y_1^{(k+1)} = y_0^{(k)} - \left(D\Phi\left(y_1^{(k)}; x_0, y_0, h\right)\right)^{-1} \Phi\left(y_1^{(k)}; x_0, y_0, h\right) \qquad k = 0, 1, 2, \ldots \tag{4}$$

1.2. Dormand & Prince.

```
function [x, y] = DOPRI5( f,x0,xN,y0,h0 )
abstol = repmat(1e-6, length(y0), 1);
reltol = repmat(1e-6, length(y0), 1);
A = [
        [0,0,0,0,0,0,0],
        [1./5,0,0,0,0,0,0],
        [3./40,9./40,0,0,0,0,0],
```

```
function [ x,y ] = ITR( f,x0,xN,y0,N,tol,maxit )
    h = ( xN - x0 )/N;
    x = x0:h:xN;
    y = zeros(length(y0), N + 1);
    y(:,1) = y0;
    for k = 1:N
        phi = @(y1) y(:,k) + h/2 * (f(x(k),y(:,k)) + f(x(k + 1), y1));
        y(:,k + 1) = fixediter( phi,y(:,k),tol );
    end
end
```

Listing 2. Implicit trapezoidal rule.

```
        [44./45,-56./15,32./9,0,0,0,0],
        [19372./6561,-25360./2187,64448./6561,-212./729,0,0,0],
        [9017./3168,-355./33,46732./5247,49./176,-5103./18656,0,0],
        [35./384,0,500./1113,125./192,-2187./6784,11./84,0]
];
b = [
        35./384,
        0.,
        500./1113,
        125./192,
        -2187./6784,
        11./84,
        0.
];
B = [
        5179./57600,
        0.,
        7571./16695,
        393./640,
        -92097./339200,
        187./2100,
        1./40
];
facmax = 2;
power = 5;
fac = (.25)^(1./power);
x = x0;
y = y0;
n = length(y0);
while x(end) < xN
    if x(end) + h0 >= xN
        h0 = xN - x(end);
    end
    c = sum(A, 2);
    k = zeros(length(y(:,end)), length(A));
    for i = 1:length(A)
        k(:,i) = f(x(end) + h0 * c(i), y(:,end) + h0 * k(:,1:i) * A(i,1:i)');
    end
    y1 = y(:,end) + h0 * k * b;
    Y1 = y(:,end) + h0 * k * B;
    sc = abstol;
```

```
    for i = 1:n
        sc(i) = sc(i) + (max(abs(y(i,end)) , abs(y1(i))) * reltol(i));
    end
    err = sqrt( 1/n * sum(((y1 - Y1)./sc).^2) );
    if err >= realmin
        r = min(facmax, max(0.1, fac * (1/err)^(1/power)) );
    else
        r = facmax;
    end
    if err <= 1
        x(end + 1) = x(end) + h0;
        y(:,end + 1) = Y1;
        h0 = h0 * r;
        facmax = 5;
    else
        h0 = h0 * r;
        facmax = 1;
    end
end
end
```

Index