Answer to Mathematical Experiment Exam 2018

Tony Xiang

May 29, 2019

1 Problem 1

```
% Picard recursion
syms x y;
f = @(x, y) y.^2 - x.^2;
x0 = 0;
y0 = 1;
phi = @(x) y0 + x .* 0;
for i = 1: 6
    fplot(phi, [0, 1], 'linewidth', 1);
    hold on;
    symphi = phi(x);
    symf = f(x, symphi);
    symnextphi = y0 + int(symf, x0, x);
    phi = matlabFunction(symnextphi);
end
hold off;
legend(['p1'; 'p2'; 'p3'; 'p4'; 'p5'; 'p6']);
set(gca, 'fontsize', 16);

% Runge-Kutta solution
options = odeset('RelTol',1e-12,'AbsTol',1e-10);
[x, y] = ode45(@(x, y)rigid(x, y), [0 1], 1, options);
figure(2);
plot(x, y, 'linewidth', 1);

function dy = rigid(x, y)
    dy = y.^2 - x.^2;
end
```

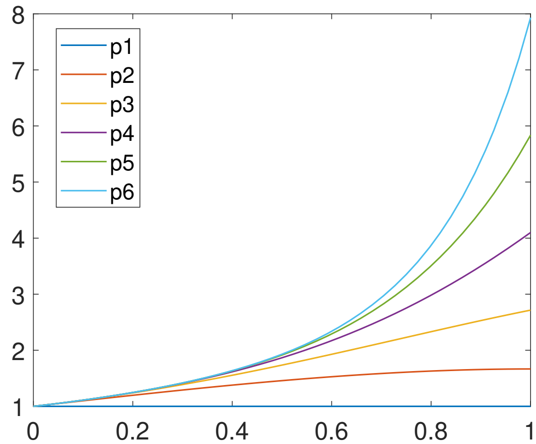

Figure 1: Picard Series

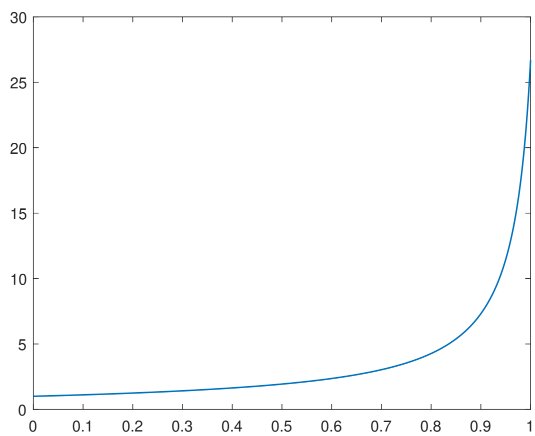

Figure 2: ODE Solution

2 Problem 2

```
f = @(x) sin(x) ./ (1 + x.^3);

lastphi = @(x) 1;
alpha = integral(@(x) sqrt(x) .* x .* lastphi(x) .* lastphi(x),
↪  0, 1) ./ integral(@(x) sqrt(x) .* lastphi(x) .* lastphi(x),
↪  0, 1);
```

```
thisphi = @(x) x - alpha;
for i = 1: 5
    alpha = integral(@(x) sqrt(x) .* x .* thisphi(x) .*
    ↪  thisphi(x), 0, 1) ./ integral(@(x) sqrt(x) .* thisphi(x)
    ↪  .* thisphi(x), 0, 1);
    beta = integral(@(x) sqrt(x) .* thisphi(x) .* thisphi(x), 0,
    ↪  1) ./ integral(@(x) sqrt(x) .* lastphi(x) .* lastphi(x),
    ↪  0, 1);
    nextphi = @(x) (x - alpha) .* thisphi(x) - beta .*
    ↪  lastphi(x);
    lastphi = @(x) thisphi(x);
    thisphi = @(x) nextphi(x);
end

syms x
sol = vpasolve(thisphi(x));
% interpretation of sol
xi = [double(sol(1)) double(sol(2)) double(sol(3))
↪  double(sol(4)) double(sol(5)) double(sol(6))];

aisol = generic_integral_coeff(6, xi);
% interpretation of aisol
ai = [double(aisol.a1) double(aisol.a2) double(aisol.a3)
↪  double(aisol.a4) double(aisol.a5) double(aisol.a6)];
result = generic_gauss_integral(f, 1, ai, xi);
result

% using integral function
answer = integral(@(x) f(x) .* sqrt(x), 0, 1);
answer

function sol = generic_integral_coeff(total_vars, xi)
    left = 0;
    right = 1;
    funct = @(x) sqrt(x);

    % constructing variables and symbols
    argument = sym('a', [1 total_vars]);

    % constucting equations
    sym_funct = sym(funct);
    for i = 1: total_vars
        key = int(sym_funct .* sym(@(x) x .^ (i - 1)), left,
        ↪  right);
        equations(i) = sum(argument .* xi .^ (i - 1)) == key;
    end
```

```
    % solving equations
    sol = vpasolve(equations);
end

function result = generic_gauss_integral(funct, density, coeff,
↪  points)
    interval = density;
    result = 0;
    for i = 1: interval
        lb_now = (i - 1) / interval;
        ub_now = lb_now + 1 / interval;
        result = result + (ub_now - lb_now) * interval_sum(@(x)
        ↪  funct(lb_now + (ub_now - lb_now) .* x), coeff,
        ↪  points);
    end
end

function interval = interval_sum(funct, coeff, points)
    interval = 0;
    for i = 1: size(coeff, 2)
        interval = interval + coeff(i) * funct(points(i));
    end
end
```

With an answer of 0.2638, compared to the MatLab built-in integral function of 0.2638.

3 Problem 3

```
t = [0 0.1000 0.2000 0.3000 0.4000 0.5000 0.6000 0.7000 0.8000
↪  0.9000 1.0000];
y = [6.9000 6.3653 6.0029 5.7894 5.7066 5.7405 5.8806 6.1194
↪  6.4522 6.8763 7.3912];
options = optimset();
options.MaxFunEvals = 10000;
options.MaxIter = 10000;
options.TolX = 1e-10;

f = @(c, t) c(1) .* exp(-c(2) .* t) + c(3) .* exp(-c(4) .* t);
[x, resnorm, residual, exitflag, output] = lsqcurvefit(f, [1 1 1
↪  1], t, y, [], [], options);
x
```

The answer is:

$$c_1 = 4.3998$$
$$\lambda_1 = 2.0001$$
$$c_2 = 2.5002$$
$$\lambda_2 = -0.9999$$

4 Problem 4

```
f = @(t) exp(-t / 2) .* (cos(t)).^4 - abs(t);
fplot(f, [-4 2], 'linewidth', 1);
hold on;
fplot(0, [-4 2]);
hold off;
x1 = fsolve(f, -3.8);
x2 = fsolve(f, -2.7);
x3 = fsolve(f, -0.5);
x4 = fsolve(f, 0.4);
x1, x2, x3, x4
```

The plot of that function is:

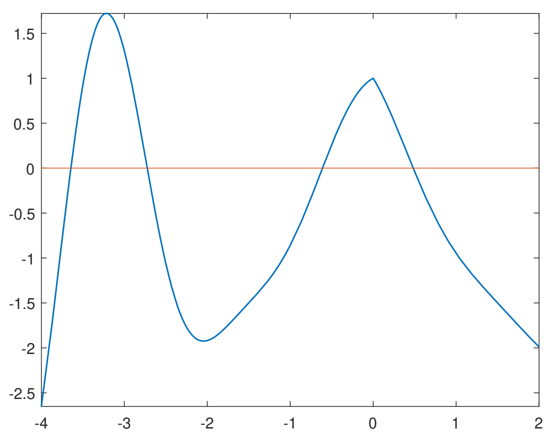

Figure 3: Plot of the function

There are 4 solutions:

$$x_1 = -3.6447$$
$$x_2 = -2.7237$$
$$x_3 = -0.6110$$
$$x_4 = 0.4831$$