

11. 使用4阶Runge-Kutta方法求解如下所示:

函数定义如下:

```
function rungekutta(f::Function, xspan, y0, num)
    a, b = xspan
    x0 = a
    h = (b - a) / num
    xs, ys = zeros(num), zeros(num)
    for n = 1:num
        k1 = h * f(x0, y0)
        k2 = h * f(x0 + h / 2, y0 + k1 / 2)
        k3 = h * f(x0 + h / 2, y0 + k2 / 2)
        k4 = h * f(x0 + h, y0 + k3)
        x1 = x0 + h
        y1 = y0 + 1 / 6 * (k1 + 2k2 + 2k3 + k4)
        xs[n], ys[n] = x0, y0 = x1, y1
    end
    xs, ys
end
```

计算结果如下:

Runge-Kutta Solver:

x	h=0.2 Pred y	h=0.1 Pred y	h=0.05 Pred y	h=0.001 Pred y
0.10000000	NaN	0.33333333	0.14062500	0.13533528
0.20000000	5.00000000	0.11111111	0.01977539	0.01831564
0.30000000	NaN	0.03703704	0.00278091	0.00247875
0.40000000	25.00000000	0.01234568	0.00039107	0.00033546
0.50000000	NaN	0.00411523	0.00005499	0.00004540
0.60000000	125.00000000	0.00137174	0.00000773	0.00000614
0.70000000	NaN	0.00045725	0.00000109	0.00000083
0.80000000	625.00000000	0.00015242	0.00000015	0.00000011
0.90000000	NaN	0.00005081	0.00000002	0.00000002
1.00000000	3125.00000000	0.00001694	0.00000000	0.00000000

图1: 左图为h=0.2时所求数值解, 右图为h=0.1时所求数值解

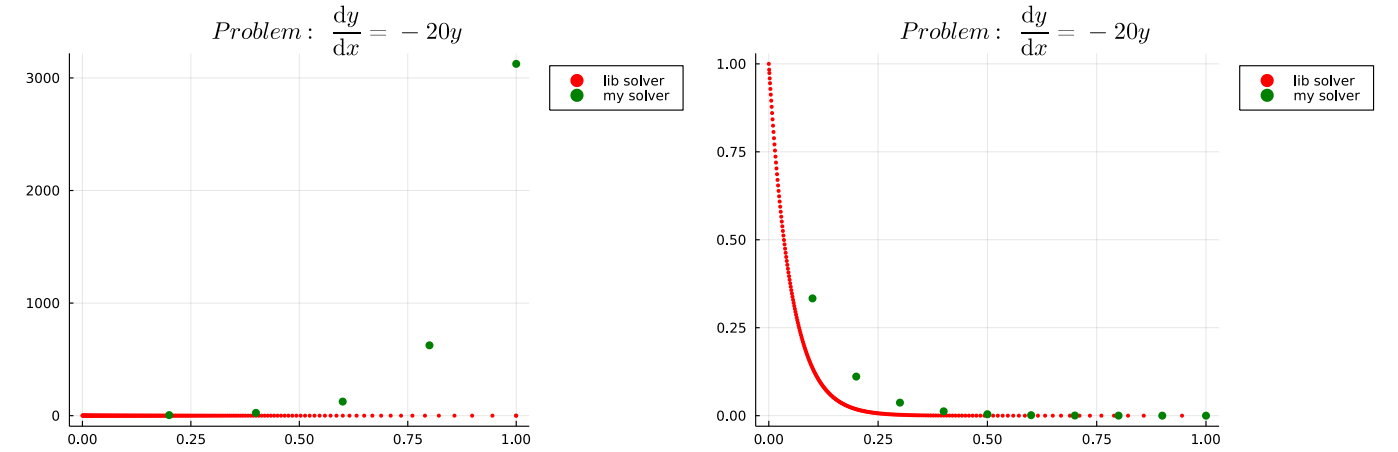
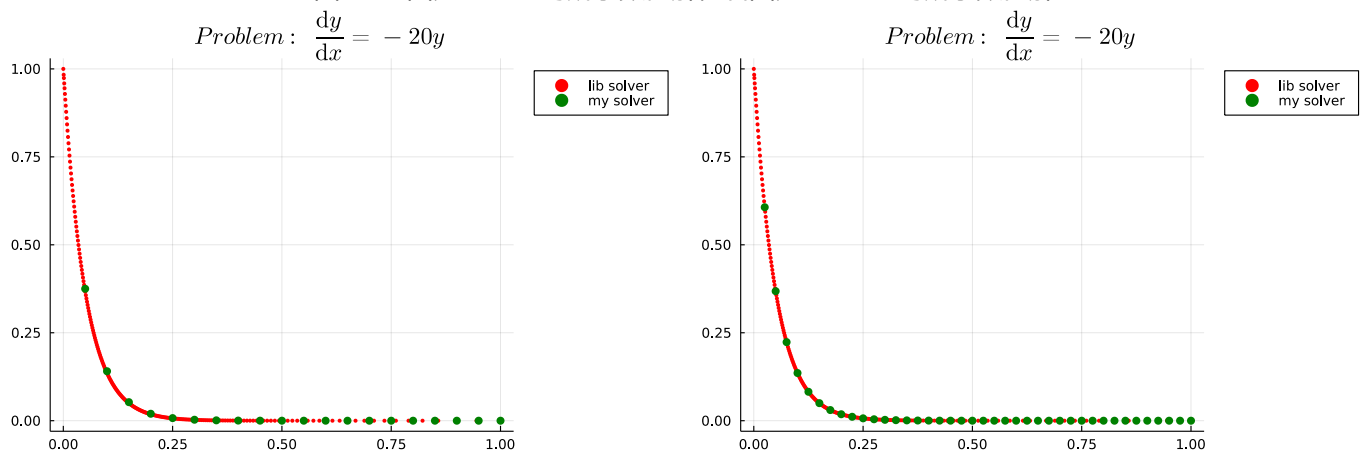


图2: 左图为h=0.05时所求数值解, 右图为h=0.025时所求数值解



具体的求解过程调用的 Julia 代码:

```
using DifferentialEquations
using Plots
using LaTeXStrings
using Statistics
using ImplicitEquations
using PrettyTables

for h in [0.2, 0.1, 0.05, 0.025]
    f(y, p, x) = -20y
    xspan = (0.0, 1.0)
    y0 = 1.0
    prob = ODEProblem(f, y0, xspan)
    alg = RK4()
    sol = solve(prob, alg, reltol=1e-8, abstol=1e-8)
    plot(title=L"~~~~~ Problem:\ \frac{\mathrm{d}}{\mathrm{d}} y\{\mathrm{d}}
x\}=-20y", legend=:outertopright)
    plot!(sol.t, sol.u, seriestype=:scatter, markersize=2, msw=0, color=:red, label="lib
solver")

    f(x, y) = -20y
    println("My Runge-Kutta Solver:")
    num = convert(Integer, 1.0 / h)
    xs, ys = rungekutta(f, xspan, y0, num)
    data = [xs ys]
    header = (["x", "Pred y"])
    pretty_table(
        data;
        alignment=[:c, :c],
        header=header,
        header_crayon=:crayon"bold",
        # tf = tf_markdown,
        formatters=ft_printf("%14.8f"))
    p = plot!(xs, ys, seriestype=:scatter, markersize=4, msw=0, color=:green, label="my
solver")
    display(p)
end
```

