Problems of Chapter 10.5

张志心 计科 2106

日期：2024 年 4 月 30 日

Exercise 10.156

Does the length of a short thick line segment in Figure 10.9 represent the one-step error in Definition 10.159? If so, prove it; otherwise derive an expression of the represented quantity.

解. $\mathcal{L}\boldsymbol{u}(t_n) = \boldsymbol{u}(t_{n+1}) - \boldsymbol{u}(t_n) - k\Phi(\boldsymbol{u}(t_n), t_n; k)$，short thick line 的长度为

$$\boldsymbol{u}(t_{n+1}) - \boldsymbol{u}(t_n) + \boldsymbol{U}^n - \boldsymbol{U}^{n+1} = \boldsymbol{u}(t_{n+1}) - \boldsymbol{u}(t_n) - k\Phi(\boldsymbol{U}^n, t_n; k).$$

所以不是 modified Euler 方法的单步误差，而是

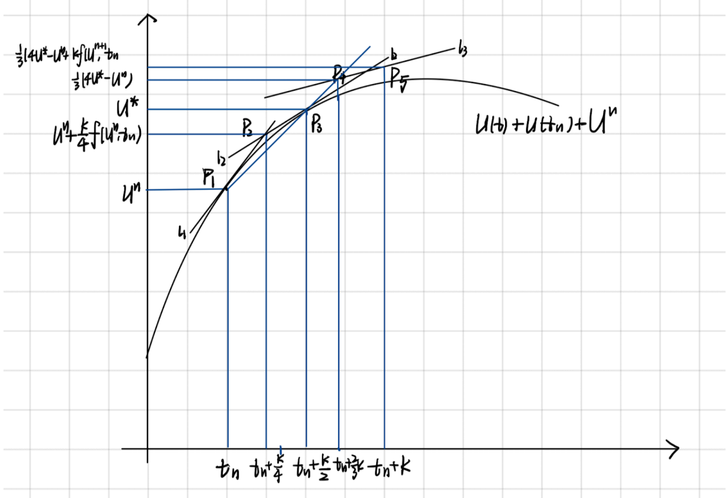
$$\mathcal{L}\boldsymbol{u}(t_n) + k(\Phi(\boldsymbol{u}(t_n), t_n; k) - \Phi(\boldsymbol{U}^n, t_n; k)).$$

□

Exercise 10.158

Give a geometric interpretation of TR-BDF2 by drawing a figure similar to those for the improved Euler method and the modified Euler method.

解.



以下几何解释中方程组的维数为 1，即 (t,u) 是二维平面上的点。给定 t_n, U^n 的情况下，U^{n+1} 将按如下步骤作出：

1. $P_1(t_n, U^n)$；
2. l_1 是过 P_1 的积分曲线在 $t = t_n$ 处的切线，它的斜率为 $f(U^n, t_n)$；
3. P_2 是 l_1 上横坐标为 $t = t_n + \frac{k}{4}$ 的点，即 $P_2(t_n + \frac{k}{4}, U^n + \frac{k}{4}f(U^n, t_n))$；
4. l_2：一条积分曲线在 $t = t_n + \frac{k}{2}$ 的切线，该切线过 P_2，它的斜率为 $f(\cdot, t_n + \frac{k}{2})$；
5. P_3：l_2 上横坐标为 $t = t_n + \frac{k}{2}$ 的点，记其纵坐标为 U^*，即 $P_3(t_n + \frac{k}{2}, U^*)$；
6. P_4：延长 P_1P_3 至横坐标为 $t = t_n + \frac{2}{3}k$，即 $P_4(t_n + \frac{2}{3}k, \frac{1}{3}(4U^* - U^n))$；
7. l_3：一条积分曲线在 $t = t_n + k$ 的切线，该切线过 P_4，它的斜率为 $f(\cdot, t_n + k)$；
8. P_5：l_3 上横坐标为 $t = t_n + k$ 的点，记其纵坐标为 U^{n+1}，即 $P_5(t_n + k, \frac{1}{3}(4U^* - U^n + kf(U^{n+1}, t_n + k)))$。

□

Exercise 10.162

Use recursive Taylor expansions to derive the k^3 term in the one-step error $\mathcal{L}\boldsymbol{u}(t_n)$ of the explicit midpoint method, verifying $\mathcal{L}\boldsymbol{u}(t_n) = \Theta(k^3)$, i.e., the explicit midpoint method is second-order accurate.

证明.

$$
\begin{aligned}
\mathcal{L}\boldsymbol{u}(t_n) &= \boldsymbol{u}(t_{n+1}) - \boldsymbol{u}(t_n) - k\Phi(\boldsymbol{u}(t_n), t_n; k) \\
&= (\boldsymbol{u}(t_n) + k\boldsymbol{u}'(t_n) + \frac{k^2}{2}\boldsymbol{u}''(t_n) + \frac{k^3}{6}\boldsymbol{u}'''(t_n) + O(k^4)) - \boldsymbol{u}(t_n) - k\boldsymbol{f}(\boldsymbol{u}(t_n) + \frac{k}{2}\boldsymbol{f}(\boldsymbol{u}(t_n), t_n), t_n + \frac{k}{2}) \\
&= k\boldsymbol{u}'(t_n) + \frac{k^2}{2}\boldsymbol{u}''(t_n) + \frac{k^3}{6}\boldsymbol{u}'''(t_n) - k\boldsymbol{f}(\boldsymbol{u}(t_n) + \frac{k}{2}\boldsymbol{u}'(t_n), t_n + \frac{k}{2}) + O(k^4) \\
&= k\boldsymbol{u}'(t_n) + \frac{k^2}{2}\boldsymbol{u}''(t_n) + \frac{k^3}{6}\boldsymbol{u}'''(t_n) - k\boldsymbol{f}(\boldsymbol{u}(t_n), t_n) - \frac{k^2}{2}\boldsymbol{f}_t(\boldsymbol{u}(t_n), t_n) - \frac{k^2}{2}\boldsymbol{u}'(t_n)\boldsymbol{f}_n(\boldsymbol{u}(t_n), t_n) \\
&\quad - \frac{k^3}{8}\boldsymbol{f}_{tt}(\boldsymbol{u}(t_n), t_n) - \frac{k^3}{8}\boldsymbol{u}'(t_n)^2\boldsymbol{f}_{uu}(\boldsymbol{u}(t_n), t_n) - \frac{k^3}{4}\boldsymbol{u}'(t_n)\boldsymbol{f}_{ut}(\boldsymbol{u}(t_n), t_n)
\end{aligned}
$$

因为

$$
\begin{aligned}
\boldsymbol{u}' &= \boldsymbol{f}, \\
\boldsymbol{u}'' &= \boldsymbol{f}_u\boldsymbol{u}' + \boldsymbol{f}_t = \boldsymbol{f}_u\boldsymbol{f} + \boldsymbol{f}_t, \\
\boldsymbol{u}''' &= \boldsymbol{f}_{uu}\boldsymbol{f}\boldsymbol{u}' + \boldsymbol{f}_{ut}\boldsymbol{f} + \boldsymbol{f}_u^2\boldsymbol{u}' + \boldsymbol{f}_u\boldsymbol{f}_t + \boldsymbol{f}_{ut}\boldsymbol{u}' + \boldsymbol{f}_{tt} = \boldsymbol{f}_{uu}\boldsymbol{f}^2 + 2\boldsymbol{f}_{ut}\boldsymbol{f} + \boldsymbol{f}_u^2\boldsymbol{f} + \boldsymbol{f}_u\boldsymbol{f}_t + \boldsymbol{f}_{tt}
\end{aligned}
$$

所以

$$
\begin{aligned}
\mathcal{L}\boldsymbol{u}(t_n) &= k(\boldsymbol{u}' - \boldsymbol{f}) + \frac{k^2}{2}(\boldsymbol{u}'' - \boldsymbol{f}_t - \boldsymbol{f}_u\boldsymbol{f}) + k^3(\frac{1}{6}\boldsymbol{u}''' - \frac{\boldsymbol{f}_{tt} + 2\boldsymbol{f}_{ut}\boldsymbol{f} + \boldsymbol{f}_{uu}\boldsymbol{f}^2}{8}) + O(k^4) \\
&= k^3(\frac{\boldsymbol{u}'''}{6} - \frac{\boldsymbol{f}_{tt} + 2\boldsymbol{f}_{ut}\boldsymbol{f} + \boldsymbol{f}_{uu}\boldsymbol{f}^2}{8}),
\end{aligned}
$$

系数为

$$
\frac{\boldsymbol{u}'''(t_n)}{6} - \frac{\boldsymbol{f}_t t(\boldsymbol{u}(t_n), t_n) + 2\boldsymbol{f}_{ut}(\boldsymbol{u}(t_n), t_n)\boldsymbol{f}(\boldsymbol{u}(t_n), t_n) + \boldsymbol{f}_{uu}(\boldsymbol{u}(t_n), t_n)\boldsymbol{f}(\boldsymbol{u}(t_n), t_n)^2}{8}.
$$

即 $\mathcal{L}\boldsymbol{u}(t_n) = O(k^3)$，所以 explicit 中点法有二阶精度。 □

Exercise 10.171

Show that the TR-BDF2 method in (10.106) satisfies

$$R(z) = \frac{1 + \frac{5}{12}z}{1 - \frac{7}{12}z + \frac{1}{12}z^2},$$

and $R(z) - e^z = O(z^3)$ as $z \to 0$.

解. 对 IVP $\boldsymbol{u}' = \lambda \boldsymbol{u}$，有

$$\begin{cases} \boldsymbol{U}^* = \boldsymbol{U}^n + \frac{k}{4}(\lambda \boldsymbol{U}^n + \lambda \boldsymbol{U}^*) \\ \boldsymbol{U}^{n+1} = \frac{1}{3}(4\boldsymbol{U}^* - \boldsymbol{U}^n + k\lambda \boldsymbol{U}^{n+1}) \end{cases}$$

$$\boldsymbol{U}^* = \boldsymbol{U}^n + \tfrac{z}{4}(\boldsymbol{U}^n + \boldsymbol{U}^*) \Rightarrow \boldsymbol{U}^* = \frac{1 + \frac{z}{4}}{1 - \frac{z}{4}}\boldsymbol{U}^n.$$

$$\boldsymbol{U}^{n+1} = \frac{4\boldsymbol{U}^* - \boldsymbol{U}^n + z\boldsymbol{U}^{n+1}}{3} \Rightarrow \boldsymbol{U}^{n+1} = \frac{4\boldsymbol{U}^* - \boldsymbol{U}^n}{3 - z} = \frac{4\frac{(4+z)}{4-z} - 1}{3 - z}\boldsymbol{U}^n = \frac{5z + 12}{(z-3)(z-4)} = \frac{1 + \frac{5}{12}z}{1 - \frac{7}{12}z + \frac{1}{12}z^2}.$$

$$\begin{aligned} R(z) &= \frac{9}{1 - \frac{z}{3}} - \frac{8}{1 - \frac{z}{4}} \\ &= 9\left(1 + \frac{z}{3} + \frac{z^2}{9} + \frac{z^3}{27} + O(z^4)\right) - 8\left(1 + \frac{z}{4} + \frac{z^2}{16} + \frac{z^3}{64} + O(z^4)\right) \\ &= 1 + z + \frac{z^2}{2} + \frac{5z^3}{24} + O(z^4) \\ &= e^z + O(z^3). \end{aligned}$$

□

Exercise 10.176

Reproduce the results in Example 10.175 and explain in your own language why the first-order backward Euler method is superior to the second-order trapezoidal method.

解. 以下是 Euler method 求解以及画图的代码：该程序计算了不同步长和 η 下的无穷范数。

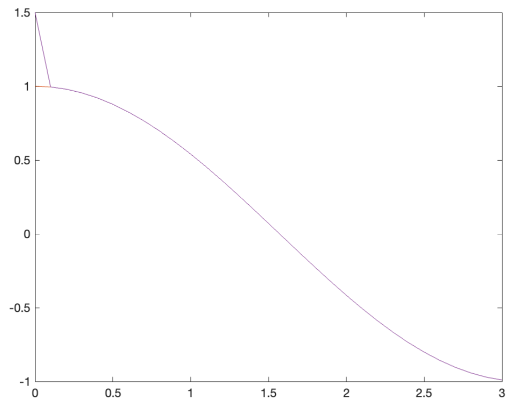
```
function backward_euler
    solve(-1e6, 3, 0.2, 1);
    figure;
    t = (0:30)/10;
    Ans = exact(-1e6, 1, t);
    plot(t, Ans);
    hold on;
    plot(t, solve(-1e6, 3, 0.1, 1));

    solve(-1e6, 3, 0.05, 1);
    solve(-1e6, 3, 0.2, 1.5);
    t = (0:30)/10;
    Ans = exact(-1e6, 1.5, t);
```

```
    plot(t, Ans);
    hold on;
    plot(t, solve(-1e6, 3, 0.1, 1.5));

    solve(-1e6, 3, 0.05, 1.5);
end

function P = exact(lmd, u0, t)
    P = exp(lmd * t) .* (u0 - 1) + cos(t);
end

function P = step(lmd, k, u, t)
    t1 = t + k;
    P = (u - k * (lmd * cos(t1) + sin(t1))) / (1 - k * lmd);
end

function P = solve(lmd, T, k, u0)
    N = floor(T / k);
    u = zeros(1, N);
    u(1) = u0;
    err = 0;
    for i = 1:N
        u(i+1) = step(lmd, k, u(i), (i-1)*k);
        exact_value = exact(lmd, u0, i*k);
        err = max(err, abs(u(i+1) - exact_value));
    end
    fprintf("k = %f, u0 = %f, err = %e\n", k, u0, err);
    P = u;
end
```

程序输出:

```
>> backward_euler
k = 0.200000, u0 = 1.000000, err = 9.900173e-08
k = 0.100000, u0 = 1.000000, err = 4.987457e-08
k = 0.050000, u0 = 1.000000, err = 2.498388e-08
k = 0.200000, u0 = 1.500000, err = 2.400986e-06
k = 0.100000, u0 = 1.500000, err = 4.950075e-06
k = 0.050000, u0 = 1.500000, err = 9.974816e-06
```

作图并与真解对比：发现曲线几乎重合，Backward-Euler 对于本问题是稳定的。



对于 trapezoidal 方法，使用了不同的 step 函数，代码如下：

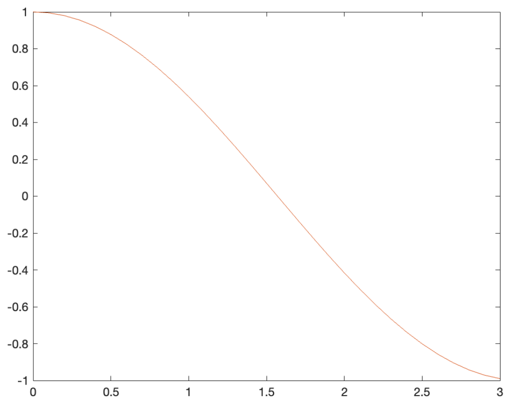
```
  function backward_euler
    solve(-1e6, 3, 0.2, 1);
    figure;
    t = (0:30)/10;
    Ans = exact(-1e6, 1, t);
    plot(t, Ans);
    hold on;
    plot(t, solve(-1e6, 3, 0.1, 1));

    solve(-1e6, 3, 0.05, 1);
    solve(-1e6, 3, 0.2, 1.5);
    t = (0:30)/10;
    Ans = exact(-1e6, 1.5, t);
    plot(t, Ans);
    hold on;
    plot(t, solve(-1e6, 3, 0.1, 1.5));

    solve(-1e6, 3, 0.05, 1.5);
end

function P = exact(lmd, u0, t)
    P = exp(lmd * t) .* (u0 - 1) + cos(t);
end

function P = step(lmd, k, u, t)
```

```
    t1 = t + k;
    P = (u - k * (lmd * cos(t1) + sin(t1))) / (1 - k * lmd);
end
```

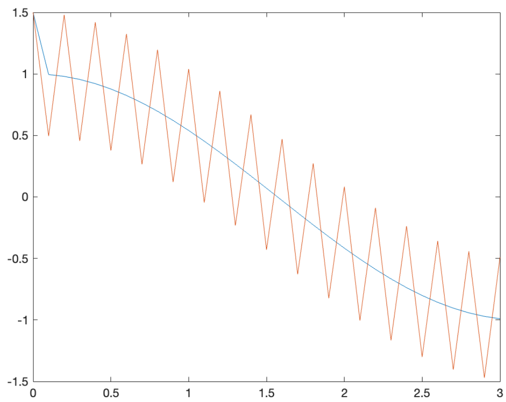
程序输出:

```
>> trapezoidal
k = 0.200000, u0 = 1.000000, err = 5.547095e-07
k = 0.100000, u0 = 1.000000, err = 5.263548e-07
k = 0.050000, u0 = 1.000000, err = 5.128314e-07
k = 0.200000, u0 = 1.500000, err = 4.999898e-01
k = 0.100000, u0 = 1.500000, err = 4.999799e-01
k = 0.050000, u0 = 1.500000, err = 4.999600e-01
```

对于 $\eta = 1$，求解误差很小，图像与真解几乎重合:



但是对于 $\eta = 1.5$，求解结果在真实解附近来回振荡并不收敛:



Backward Euler 方法是 L 稳定的，但是 trapezoidal 方法不是，因此前者在求解特征值很大的问题时对于初值的敏感程度不会过大，收敛效果更好。 □