화공전산

2020 숙제 4

11/3 (화) 제출 마감 (종료와 차단 시간은 클래스룸에서 확인하세요.)

주의

- 풀이 과정과 프로그래밍 문제를 포함한 모든 문제의 답안을 정리하여 `B123456hw40.pdf`로 제출하시오.
- 전체 36점 만점이고 각 문제 동일 배점. 문제 별 부문제 동일 배점.
- 풀이의 타당성과 설득력 > 답의 정확성
- 제출할 파일은 `B123456hw4.pdf` 1 개와 `.py` 3 개이다.

문제

4.1 초기값 문제 (1)의 수치해를 구하자.

$$(1-x^2)y''(x) - 2xy'(x) + 6y(x) = 0 \qquad y(0) = 1 \qquad y'(0) = 0 \tag{1}$$

(a) 파이선 스크립트를 작성하시오. `solve_ivp`에서 보폭이 0.1인 `t_eval`을 사용하시오. (`B123456hw41.py`)

```
...
from scipy.integrate import solve_ivp
r = solve_ivp(...)
from matplotlib import pyplot as plt
...
plt.show()
```

(b) 구간 $0 \le x \le x_e$에서 초기값 문제를 위의 `.py`로 풀고 답하시오.

- $x_e = 0.9$일 때 $y(x_e)$를 출력하고 $y(x)$의 그래프를 그리시오.
- $x_e = 1.1$일 때 $y(x_e)$를 출력하고 $y(x)$의 그래프를 그리시오.
- $x_e = 1$일 때 $y(x)$의 그래프를 그리시오.

(c) 이 문제의 특이점에 대하여 논하시오.

다음을 참고 또는 적용하시오.

- 각 x_e에 대하여 `r.t`를 자세히 살펴보시오.
- 초기값 문제 (1)의 엄밀해 $y_{\text{exact}}(x) = 1 - 3x^2$이다.
- 그래프를 그릴 때
  - 먼저 $y_{\text{exact}}(x)$를 매끄러운 빨간 선 (—, `'r'`)으로 그리고
  - 그 다음에 수치해를 까만 ● (`'ko'`) 또는 ○ (`'ko', markerfacecolor = 'w'`)로 그리시오.
  - `**kwarg`인 `markerfacecolr`는 `mfc`로 줄여 쓸 수 있다.

4.2 H. H. Robertson은 그의 저서 *Numerical analysis: an introduction*에서 “The solution of a set of reaction rate equations”라는 글을 통하여 화학 반응의 수치 모사는 수치해법의 성능보다 화학 반응에 대한 지식이 더 중요할 수 있음을 보였다. Robertson은 다음의 예를 들어 설명하였다.

$$A \rightleftharpoons B \rightarrow C$$

반응 혼합물 중 A, B, C의 농도를 각각 C_A, C_B, C_C라 하면 각 반응의 속도는 다음과 같다.

- $A \rightarrow B$: $r_1 = k_1 C_A$
- $B \rightarrow A$: $r_2 = k_2 C_B C_C$
- $B \rightarrow C$: $r_3 = k_3 {C_B}^2$

단, $k_1 = 0.04$, $k_2 = 1 \times 10^4$, $k_3 = 3 \times 10^7$이다. 각 물질의 농도 변화는 다음 물질 수지식으로 표현된다.

$$\frac{dC_A}{dt} = -r_1 + r_2 \tag{2}$$
$$\frac{dC_B}{dt} = r_1 - r_2 - r_3 \tag{3}$$
$$\frac{dC_C}{dt} = r_3 \tag{4}$$

초기 조건은 $C_A(0) = 1$, $C_B(0) = 0$, $C_C(0) = 0$이다. 구간 $0 \le t \le 40$에서 `scipy.integrate.solve_ivp`를 이용하여 이 초기값 문제를 풀자.

(a) 파이선 스크립트를 작성하시오. (`B123456hw42.py`)

```
1  import datetime
2  ...
3  from scipy.integrate import solve_ivp
4  ...
5  print(datetime.datetime.now())
6  rb = solve_ivp(...)
7  print(datetime.datetime.now())
8  print(rb)
9  ...
10 from matplitlib import pyplot as plt
11 ...
12 plt.figure(num = 1)
13 plt.plot(r.t, r.y[0], ...)
14 ...
15 plt.figure(num = 2)
16 plt.plot(r.t, r.y[1], ...)
17 ...
18 plt.show()
```

(b) 이 스크립트에서 (t, C_A)의 그래프, (t, C_B)의 그래프, (t, C_C)의 그래프를 그리시오.

다음을 참고 또는 적용하시오.

- 6번 줄에서 `t_eval`을 사용하지 말고, 미방을 푸는 보폭의 결정을 `solve_ivp`에게 맡기자.
- 8번 줄에서 `rb.nfev`는 미방을 풀기 위하여 $d\vec{y}/dt = \vec{f}(t, \vec{y})$를 호출한 횟수이며 계산의 부하 (computational load)를 나타낸다.
- 1, 5, 7 번 줄을 통하여 미방 푸는 시간을 측정할 수 있다.
  - 컴퓨터의 성능을 비롯한 여러가지 요인으로 그 값은 담당 교수가 얻은 값과 다를 것이며, `.py`를 반복 실행하면 매번 조금씩 다른 값이 나올 수 있다.
- 12, 15번 줄과 같이 `plt.figure`에서 `num`에 서로 다른 값을 지정하여 여러개의 그래프를 생성할 수 있다.

4.3 앞의 문제에서

- C_A나 C_C는 대략 0부터 $C_A(0)$ 사이의 값을 갖지만, $C_B \ll C_A(0)$임을 알 수 있다. 따라서 $t > 0$의 대부분의 구간에서 다음 관계가 성립됨을 짐작할 수 있다.

$$\left|\frac{dC_B}{dt}\right| \ll \left|\frac{dC_A}{dt}\right| \quad \text{and} \quad \left|\frac{dC_B}{dt}\right| \ll \left|\frac{dC_C}{dt}\right| \tag{5}$$

- `rb.nfev`로부터 보폭(stepsize)가 매우 작은 것을 알 수 있다.
- 보폭이 매우 작음에도 불구하고 (t, C_B)의 그래프는 C_B 계산에 어려움이 있음을 보인다. (그런가?)

B는 반응이 $A \to C$로 가는 중간 물질로 매우 작은 양이지만, 반응 속도를 조절한다.

준-정상 상태 근사 (5) 식에 근거하여

$$\frac{dC_B}{dt} = 0 \tag{6}$$

이라 가정하자. 이 가정을 (3) 식에 적용하면 미분 방정식 대신

$$C_B = \psi(C_A, C_C) \geq 0 \tag{7}$$

의 함수 형태로 구할 수 있다. 그런데 (6) 식과 (7) 식 사이에는 한 가지 모순된 점이 있다.

> (6) 식에 따르면 C_B는 시간에 따라 변하지 않는 상수인데 (7) 식에 따르면 C_A와 C_C가 시간에 따라 변하므로 C_B도 시간에 따라 변한다.

다만 C_B는 시간에 따라 변하더라도 (5) 식이 만족된다면 '거의' 정상상태라 할 수 있다. 이와 같은 이유로 화학 반응의 모델 식에서 (3) 식 대신 (7) 식을 사용하는 것을 준-정상 상태 근사 (quasi-steady state approximation)라 한다. 준-정상 상태 근사를 위한 미분 방정식 모델 $d\vec{z}/dt = \vec{g}(t, \vec{z})$은 다음과 같다.

- C_A와 C_C로부터 C_B를 구한다.
- 반응 속도 r_1, r_2, r_3를 구한다.
- 도함수 dC_A/dt와 dC_C/dt를 구한다.

문제 4.2의 $\vec{f}(t, \vec{y})$와 $\vec{g}(t, \vec{z})$가 다른 점은 $\vec{f}$와 $\vec{y}$는 3차원 벡터이지만, $\vec{g}$와 $\vec{z}$는 C_B가 빠진 2차원 벡터라는 점이다. 이제 새로운 초기값 문제를 풀어보자.

(a) (7) 식의 ψ를 구하시오.

(b) 파이선 스크립트를 작성하시오. (`B123456hw43.py`)

```
1  import datetime
2  ...
3  from scipy.integrate import solve_ivp
4  ...
5  print(datetime.datetime.now())
6  rc = solve_ivp(...)
7  print(datetime.datetime.now())
8  print(rc)
9  rb = solve_ivp(...)
10 ...
11 plt.show()
```

(c) 두 모델에서 구한 C_A, C_B, C_C에 대하여 (t, C_A)의 그래프, (t, C_B)의 그래프, (t, C_C)의 그래프를 그려 비교하시오.

(d) 두 모델에 대하여 `nfev` 값과 계산 시간을 비교하시오. 또 `nfev` 값과 계산 시간은 비례 관계를 가진다 할 수 있는지 판단하시오.

다음을 참고 또는 적용하시오.

- 6번 줄은 준-정상상태 모델을 푸는 줄이다. 이번에도 6번 줄에서 `t_eval`을 사용하지 말고, 미방을 푸는 보폭의 결정을 `solve_ivp`에게 맡기자.
- 원래 미방 (2) ~ (4)도 여기에서 다시 풀어서 (9번 줄) 그래프는 두 가지 모델 즉 (2), (3), (4) 식 모델과 (2), (4), (7) 식 모델의 결과를 함께 그리되 (2), (4), (7) 식 모델의 결과가 잘 구별되어 보이도록 신경쓰세요.