

수치중력 및 중력파 2024년 겨울학교 중력파 자료분석 문제

모든 문제는 igwn-py39 conda 환경에서 풀이를 진행합니다. 본인이 접속한 서버의 프롬프트에 (igwn-py39) 헤더가 보여야 정상적인 문제풀이 환경입니다. 본인의 컴퓨터가 Ubuntu 등 유닉스 계열인 경우에는 igwn-py39 conda 환경을 설치할 수 있습니다.

conda igwn-py39 환경 설치

참고 사이트([igwn-py39 - IGWN Conda Distribution \(ligo.org\)](https://ligo.org/igwn-py39-IGWN-Conda-Distribution)) 에 접속하여 설명대로 진행하면 됩니다.

1-1 본인의 운영체제(OS)에 맞는 설치 파일(igwn-py39.yaml) 다운로드

[linux-64](https://computing.docs.ligo.org/conda/environments/linux-64/igwn-py39.yaml)(<https://computing.docs.ligo.org/conda/environments/linux-64/igwn-py39.yaml>)

[osx-64](https://computing.docs.ligo.org/conda/environments/osx-64/igwn-py39.yaml)(<https://computing.docs.ligo.org/conda/environments/osx-64/igwn-py39.yaml>)

[osx-arm64](https://computing.docs.ligo.org/conda/environments/osx-arm64/igwn-py39.yaml)(<https://computing.docs.ligo.org/conda/environments/osx-arm64/igwn-py39.yaml>)

[win-64](https://computing.docs.ligo.org/conda/environments/win-64/igwn-py39.yaml)(<https://computing.docs.ligo.org/conda/environments/win-64/igwn-py39.yaml>)

1-2 igwn-py39 conda 환경 생성

```
conda env create --file igwn-py39.yaml
```

1-3 igwn-py39 conda 환경 활성화

```
conda activate igwn-py39
```

그러나 IBS에서 제공하는 서버에서 문제풀이를 하는 경우에는 위의 환경설치를 할 필요는 없습니다. 이미 IBS서버에 필요한 환경설정을 마친 상태입니다. IBS서버에서는 Jupyter 또는 IPython console 환경에서 계산하면 됩니다. 현재 IBS 시스템의 설정으로 인하여 중력파 신호를 저장한 *.gwf 파일을 읽기 위해서는 gwpy 라이브러리의 TimeSeries 클래스를 사용해야 한다. 이와 관련한 내용은 문서 [GWpy 3.0.8 documentation](#) 의 설명과 예제를 참고하고, 중력파 검출과 관련한 라이브러리는 PyCBC 이다. PyCBC와 관련한 설명과 예제는

<https://github.com/gwastro/pycbc> 와 [PyCBC: Powering Gravitational-wave Astronomy — PyCBC 2.4.dev1 documentation](#) 를 참조하십시오.

정답 제출은 <http://olaf2.ibs.re.kr:2414> 에 접속하여 정답 json 파일과 소스를 압축한 파일 두 개를 업로드 하면 된다. 서버는 사정에 따라 변경될 수 있습니다.

2024 수치상대론 및 중력파 겨울학교 정답제출

[중력파 자료분석 정답파일 보내기](#)

[수치상대론 정답파일 보내기](#)

2024 수치상대론 및 중력파 겨울학교 중력파 자료분석 정답제출

정답파일 보내기

정답파일 : 049.json

소스파일 : 006.tar.gz

2024 수치상대론 및 중력파 겨울학교 중력파 자료분석 정답제출 결과

팀 049 팀원 여러분, 축하합니다! 중력파 자료분석 정답이 정상적으로 업로드 되었습니다.

현재 제출한 정답에 대한 자동 채점의 결과는 다음과 같습니다.

각 문제에서 완전히 맞추면 정답 아니면 오답으로 표시됩니다.

팀명	049	등수	결정안됨	점수	100	문제점수합	--	도전문제	--
1번문제	오답	2번문제	오답	3번문제	정답	4번문제	오답	도전문제	정답

[홈페이지 다시하기](#)

← ↻ ⚠ 안전하지 않음 | olaf2.ibs.re.kr:2414/nr_home.htm

2024 수치상대론 및 중력파 겨울학교 수치중력 정답제출

정답파일 보내기

정답파일 : 006_nr.json 소스파일 : 006.tar.gz

[채점기준]

모든 문제의 정답은 정답을 정확하게 맞춘 경우(소수 4째 자리에서 반올림한 3째자리 까지의 수가 일치하는 경우, 답이 0.001보다 적은 경우에는 유효자리수 4자리값 일치)는 만점을 그 이외의 경우에는 정답과의 상대오차 크기에 따라 수식 (만점 - [상대오차/10] -1)으로 계산하고 음수가 나오면 0점으로 처리한다. 각 문제의 만점은 다음과 같다.

```
{
  "problem1": {
    "TaylorF2":10,
    "IMRPhenomD":10
  },
  "problem2": {
    "maximum": 10,
    "position": 10
  },
  "problem3": {
    "m1": 10,
    "m2": 10,
    "D": 10
  },
  "problem4": {
    "m1": 5,
    "m1_std": 5,
    "a1z": 5,
    "m2": 5,
    "m2_std": 5,
    "a2z": 5
  },
  "problem5": {
    "바르게 찾은 신호당 10점
    틀린 신호를 제출하면 -15점
  }
}
```

총점은 1,2,3,4번 문제의 합과 5번 문제의 경우에 제출한 답이 맞으면 10점 틀린 답을 제출한 경우에는 -15점으로 계산한다. 따라서 총점은 100점 + 추가점수로 계산하고 100점이 넘을 수도 있다. 제출하는 정답 json 파일의 형식은 다음과 같아야 한다.

```

{
  "team": "0nn",
  "password": "password for team",
  "problem1": {
    "TaylorF2": value,
    "IMRPhenomD": value
  },
  "problem2": {
    "maximum": value,
    "position": value
  },
  "problem3": {
    "m1": value,
    "m2": value,
    "D": value
  },
  "problem4": {
    "m1": value,
    "m1_std": value,
    "a1z": value,
    "m2": value,
    "m2_std": value,
    "a2z": value5
  },
  "problem5": {
    "count": k,
    "signal1": {"time":value, "m1":value, "m2":value},
    "signal2": {"time":value, "m1":value, "m2":value},
    "signal3": {"time":value, "m1":value, "m2":value},
    ...
    "signalk": {"time":value, "m1":value, "m2":value}
  }
}

```

본인의 팀명 기록

팀의 암호 기록

JSON 파일 각 항목의 필드명은 반드시 예제에 있는 것과 같이 적어야 하고 답을 제출하지 않는 문제의 경우에는 null 값을 주면 된다.

problem2: null,
problem3: null,
problem4: null,
problem5: null,

찾은 신호의 개수

찾은 신호의 개수 만큼 기록

[문제 1번] 중력파형의 match 계산 코드 작성[20점]

중력파는 매우 미세한 신호이기 때문에 특별한 방법을 사용해야 검출할 수 있다. 실제로 신호는 잡음보다 작은 크기로 나타나기 때문에 잡음 속에서 미세한 신호를 검출하기 위해서는 신호의 모양을 알아야 가능하다. 검출하는 중력파 신호의 파형을 아는 경우에는 정합필터 (Matched Filter) 기법을 사용하여 검출할 수 있다. 그러나 천체물리학적 시스템이 만들어 내는 중력파 신호를 정확하게 아는 것은 현재로서는 불가능하다. 따라서 목표로 하는 천체물리학적 시스템에 따라서 시스템을 기술하는 이론을 적용하여 근사적인 파형을 구하여 사용하게 된다. 현재 LIGO/Virgo/KAGRA collaboration(igwn-py39환경)에서 검증되고 사용되는 파형은 시간함수 또는 주파수함수 공히 72개이다. 어떤 파형이 존재하는 가를 알아보려면 다음과 같은 Python 코드를 사용하면 된다.

```

1 from pycbc.waveform import td_approximants, fd_approximants
2
3 # List of td approximants that are available
4 print("Total number of TD waveforms : {}".format(len(td_approximants())))
5 print(td_approximants())
6
7 # List of fd approximants that are currently available
8 print("Total number of FD waveforms : {}".format(len(fd_approximants())))
9 print(fd_approximants())

```

이 문제에서는 이러한 다양한 파형들 사이의 차이를 계산하는 코드를 작성하고 결과를 계산하는 문제이다. `pycbc.psd.AdvDesignSensitivityT1800044`로 psd를 만들어 사용한다. 두 파형사이의 match는 다음 수식으로 계산한다.

$$\text{Match} = \max_t \frac{\langle h_1(t) | h_2 \rangle}{\sqrt{\langle h_1 | h_1 \rangle \langle h_2 | h_2 \rangle}} \quad (1)$$

위 식에서

$$\langle a | b \rangle \equiv 4\Re \int_0^\infty \frac{a^* b}{S_n(f)} df \quad (2)$$

이고 $S_n(f)$ 는 검출기의 PSD이다. $h_1(t)$ 는 시간을 t 만큼 이동 시킨 신호의 Fourier 변환을 의미한다. 시간을 t 만큼 이동한 신호의 Fourier 변환은

$$h_1(t) = h_1(0)e^{-2\pi i f t} \quad (3)$$

로 주어진다. 이식은 간단히 Fourier변환의 정의로부터 확인할 수 있다. 그러므로 $\langle h_1(t) | h_2 \rangle$ 는 $\langle h_1(0) | h_2 \rangle$ 의 역푸리에 변환이 된다. 이 사실을 이용하면 두 파형 사이의 match는 주파수 도메인에서 두 파형의 값을 구한 후에 각 주파수에서 $\frac{a^* b}{S_n(f)}$ 값을 계산한 후에 이를 역푸리에 변환한 후 최대 값을 찾은 후에 $\sqrt{\langle h_1 | h_1 \rangle \langle h_2 | h_2 \rangle}$ 으로 나누고 알맞은 정규화 인자를 곱하여 계산하면 방정식 (1)의 값을 구할 수 있다.

주어진 질량 $m_1=10$, $m_2=8$, `sample_rate=4096`, `segment_length = 8`, `f_low = 10`, `f_high = 2048`의 조건으로 `IMRPhenomXPHM`과 `TaylorF2`, `IMRPhenomXPHM`과 `IMRPhenomD` 사이의 match 값을 계산하여 제출하시오. 사용하는 파형은 plus polarization 파형만 사용하시오. 단, 제출할 때는 다음의 함수를 작성한 소스를 다른 소스와 함께 `tar.gz` 또는 `zip`으로 압축하여 첨부 하시오.

```
def match_new(h1, h2, psd, f_low, f_high):
    # match를 계산하여 maxsnr 에 저장하는 코드
    maxsnr = 0
    return maxsnr
```

문제를 해결하기 위해서는 다음의 라이브러리가 필요하다.

```
import warnings
warnings.filterwarnings("ignore", "Wswiglal-redirect-stdio")
import numpy as np
from pycbc.waveform import fd_approximants
from pycbc.waveform import get_fd_waveform
from pycbc.psd import aLIGODesignSensitivityT1800044
from pycbc.fft import ifft
from pycbc.types import zeros, TimeSeries
from pycbc.filter import sigmasq
```

[문제 2번] 알려진 중력파 신호의 SNR값과 위치 찾기[20점]

주어진 중력파 자료 파일(H1_GW_NR2024_PROBLEM2.gwf)에 채널명 H1:NR2024_WINTER으로 저장된 중력파 신호 파일에서 신호가 갖고 있는 SNR값과 신호의 위치를 찾아 내는 것이다. 주어진 중력파 자료 파일에는 질량이 각각 (40, 25)인 IMRPhenomD를 따르는 중력파가 숨겨져 있다. 이 중력파가 갖는 SNR값과 위치를 IMRPhenomD 템플릿을 이용하여 결정하여 제출한다. 이를 알아내기 위해서는 PyCBC의 pycbc.filter의 matched_filter() 함수와, pycbc.waveform의 get_fd_waveform의 사용법을 알아야 한다. 이에 대한 자세한 설명은 PyCBC 문서 특히 [Signal Processing with GW150914 — PyCBC 2.4.dev1 documentation](#)의 내용을 참고 하시오. 주어진 gwf파일을 읽어들이기 위해서는 gwpy.timeseries의 TimeSeries를 사용해야 하며, pycbc의 다양한 함수를 사용하려면 이를 pycbc.types.timeseries의 TimeSeries로 변환해서 사용해야 한다. 점수는 찾아낸 SNR 값과 최대 SNR 값을 주는 위치 값 각각에 대하여 10점을 부여한다.

[문제 3번] 주어진 중력파 신호에서 신호가 갖는 m_1 , m_2 , D 값 결정[30점]

주어진 중력파 자료 파일(H1_GW_NR2024_PROBLEM3.gwf)에 채널명 H1:NR2024_WINTER으로 저장된 중력파 신호 파일에서 신호가 갖고 있는 질량 m_1 , m_2 와 신호원까지의 거리 D 를 결정하여 제출한다. 이를 알아내기 위해서는 PyCBC의 pycbc.filter의 matched_filter() 함수와, pycbc.waveform의 get_fd_waveform의 사용법을 알아야 한다. 이에 대한 자세한 설명은 PyCBC 문서 특히 [Signal Processing with GW150914 — PyCBC 2.4.dev1 documentation](#)의 내용을 참고 하시오. 점수는 찾아낸 m_1 , m_2 , D 값 각각에 대하여 10점을 부여한다. 신호가

갖고 있는 질량을 알지 못하기 때문에 모든 가능한 값을 이용하여 최적의 신호크기를 주는 값을 결정해야 한다. LALSuite에서는 이를 위하여 Template bank search 라는 효율적인 방법을 사용하고 있다. 여기서는 m_1, m_2 값을 10부터 60까지 1간격으로 모든 값에 대하여 가장 큰 SNR 값을 주는 질량 값을 찾으면 된다. ($m_2 \leq m_1$ 이라는 것에 주의할 것) 이렇게 하면 검출기의 antenna pattern 등을 고려하지 않기 때문에 정확하게 질량 값을 정할 수는 없지만 신호가 있는 위치와 대강의 질량은 결정할 수 있다. 최적의 SNR 값을 찾았으면 이 신호원까지의 거리는 사용한 template이 g 라면 $D = 2 \sqrt{\langle g|g \rangle} / \text{SNR}$ 로 구할 수 있다. 물론 이 값도 근사적인 값이다. 이와 더불어 문제 2에서 구한 것과 같이 신호가 도착한 시간을 구하는 것이 좋다. 파형 template은 IMRPhenomD를 사용하시오.

[문제4] 주어진 중력파 자료파일에서 bilby를 이용한 모수추정[30점]

주어진 중력파 자료 파일(H1_GW_NR2024_PROBLEM3.gwf, L1_GW_NR2024_PROBLEM3.gwf)에 채널명 각각 H1:NR2024_WINTER, L1:NR2024_WINTER 으로 저장된 중력파 신호 파일에서 신호가 갖고 있는 질량 m_1, m_2, a_{1z}, a_{2z} , 그리고 m_1, m_2 에 대한 표준편차를 계산하여 제출한다. 문제를 풀기 위해서는 <https://github.com/gw-odw/odw-2023>를 참조한다. 특히 bilby를 활용한 모수추정을 설명한 [odw-2023/Tutorials/Day_3/Tuto_3.2_Parameter_estimation_for_compact_object_mergers.ipynb at main · gw-odw/odw-2023 · GitHub](https://github.com/gw-odw/odw-2023/blob/main/Tutorials/Day_3/Tuto_3.2_Parameter_estimation_for_compact_object_mergers.ipynb) 를 참고한다. 참고 사이트와의 차이는 우리는 저장되어 있는 gwf 파일을 사용하고 psd도 pycbc.psd.AdvDesignSensitivityT1800044 를 사용한다는 점이다. 그리고 신호의 도착시간은 문제 3에서 찾은 값을 사용한다. 우리가 모수추정할 값은 m_1, m_2, a_{1z}, a_{2z} 이지만 실제로 모수 추정을 위한 posterior sample을 생성할 때는 거리 D 까지 같이 해야 한다. 생성된 posterior sample을 이용하여 m_1, m_2 에 대한 corner plot을 그리면 m_1, m_2 의 값과 표준 편차를 알 수 있다. 상한과 하한이 다를 경우에는 하한 값을 제출한다. a_{1z}, a_{2z} 의 값은 a_1, a_2 에 대한 corner plot을 그려서 그 값을 알 수 있다. 각 항목 값마다 5점씩 부여한다. **표준편차의 값은 실제로 90% confidence interval을 계산한 후에 중간 값에서 하한 값을 뺀 값을 제출한다. 계산한 posterior sampler에서 중간값(median)과 90% confidence interval을 계산하는 예제는 위에 언급한 사이트를 참고하시오.**

[문제 5] 주어진 중력파 자료 파일에서 중력파 신호를 찾기(time, m_1, m_2)[가감점]

주어진 중력파 자료 파일(H1_GW_NR2024_PROBLEM5.gwf)에 채널명 H1:NR2024_WINTER

으로 저장된 중력파 신호 파일에서 신호가 갖고 있는 질량 m_1 , m_2 와 신호의 도착시간을 결정하여 제출한다. 이를 알아내기 위해서는 PyCBC의 `pycbc.filter`의 `matched_filter()`함수와, `pycbc.waveform`의 `get_fd_waveform`의 사용법을 알아야 한다. 이에 대한 자세한 설명은 PyCBC문서 특히 [Signal Processing with GW150914 — PyCBC 2.4.dev1 documentation](#)의 내용을 참고 하시오. 신호들은 서로 2초 이상 떨어져 있고 질량 $m_1=45$, $m_2=30$ 로 SNR을 계산할 때 $SNR>8$ 이라고 가정한다. 점수는 찾아낸 신호에 대하여 m_1 , m_2 , time 값을 10%오차 내에서 찾았으면 찾은 각 신호에 대하여 10점을 부여한다. 그러나 찾았다고 제출한 신호의 값이 이 범위를 벗어나면 감점 15점을 부여한다.