

令和4年度修士論文  
流体シミュレーションを用いた重力波望遠鏡 KAGRA の  
地下水がもたらす重力勾配雑音の推定

東京工業大学大学院理学院物理学系 宗宮研究室  
21M00487 鈴木孝典

2023年3月16日

---

## Abstract

Dr Einstein's General Theory of Relativity (1916) showed that space is distorted around an object of mass. It also predicted the existence of "gravitational waves", a physical phenomenon in which the distortion of space is transmitted like waves as the object moves. Since the existence of gravitational waves was predicted in general relativity, various attempts have been made to observe them. Currently, the most common method of observing gravitational waves is through gravitational wave telescopes using Michelson interferometers. To explain the detection mechanism, when a spatial distortion caused by gravitational waves passes through the Michelson interferometer, a difference length of two light paths that should be the same distance is created. This length affects the output of the Michelson interferometer, therefore confirming that gravitational waves have passed through. In fact, several countries have built and are operating large gravitational wave telescopes using Michelson interferometers. Among them, the US gravitational wave telescope LIGO detected gravitational waves for the first time on 14 September 2015. This observation came about 100 years after the existence of gravitational waves was predicted in general relativity. Attempts to observe gravitational waves continue with projects such as LISA, a third-generation gravitational wave telescope to be built in space, and the Einstein Telescope, which aims to further improve sensitivity.

In Japan, KAGRA, a large low-temperature gravitational wave telescope, was built in Kamioka(Hida City, Gifu Prefecture), and has participated in observations together with other gravitational wave telescopes. The distortion of space caused by gravitational waves is very small, so highly sensitive observations are needed to detect gravitational waves. In KAGRA, various efforts have been made to reduce noise, such as building underground to reduce ground noise and using mirrors with low temperatures to reduce thermal noise.

This Master's thesis focuses on the gravity gradient noise(Newtonian noise) caused by water in KAGRA's noise reduction efforts. KAGRA drains this groundwater through pipes. Fluctuations in the mass of the water in the pipes cause local gravitational fluctuations, which in turn cause the mirror to shake (fluctuations in the direct gravitational interaction with the water). This is the gravity gradient noise(Newtonian noise) of the water. Furthermore, the closer the distance between the pipe and the mirror, the greater this effect will be and it can become noise to the KAGRA design sensitivity. The purpose of this Master's thesis is to estimate the Newtonian noise from water in KAGRA and to evaluate how it affects KAGRA. Newtonian noise is difficult to measure directly, and previous studies have estimated Newtonian noise by theoretically predicting the water surface. This thesis attempts to estimate the magnitude of Newtonian noise by reproducing water in a pipe using fluid simulation. As the aim was to estimate the Newtonian noise of KAGRA, a simulation was carried out that reproduced the KAGRA pipe. In addition, a simple straight pipe was designed and simulated to see how the Newtonian noise of water varies with the flow rate and the roughness of the pipe. In addition, since the Einstein Telescope will be constructed underground, the Newtonian noise from groundwater may affect the design sensitivity of the ET, which is an assumption situation. The results of these simulations show that the magnitude of the Newtonian noise is smaller than the KAGRA design sensitivity under all conditions and does not affect the KAGRA design sensitivity.

## はじめに

Einstein 博士の一般相対性理論 (1916 年) の中では質量を持つ物体の周りでは空間がゆがむことが示されている。さらに、その物体が動くことで空間のゆがみが波のように伝わる物理現象、「重力波」の存在が予言されていた。一般相対性理論の中で重力波の存在が予言されて以降、重力波を観測しようと様々な試みが今日にいたるまでされてきた。現在ではマイケルソン干渉計を利用した重力波望遠鏡による重力波観測の方法が主流である。簡単に検出の仕組みを説明すると重力波による空間のゆがみがマイケルソン干渉計を通過すると、2 つの同距離であるはずの光路の長さに差が生じ、マイケルソン干渉計の出力に影響が現れるため、それを見ることで重力波が通ったことを確認できる。実際にいくつかの国でマイケルソン干渉計を利用した大型の重力波望遠鏡が建設され、稼働している。その中で、アメリカの重力波望遠鏡 LIGO が 2015 年 9 月 14 日に重力波を初検出した。これは一般相対性理論の中で重力波の存在が予言されてから約 100 年後の観測である。さらに、第三代重力波望遠鏡として宇宙に建設する LISA や、さらなる感度向上を目指した Einstein Telescope などのプロジェクトが進んでおり、重力波を観測しようとする試みは歩みを進め続けている。

日本においても大型低温重力波望遠鏡 KAGRA が岐阜県飛騨市神岡町に建設され、他の重力波望遠鏡とともに観測に参加した実績を持つ。KAGRA もマイケルソン干渉計を利用した重力波望遠鏡であり、サファイア鏡を用いてマイケルソン干渉計を構築している。重力波による空間のゆがみはとても小さいものであり、重力波検出には高感度の観測が要求される。雑音をなるべく減らすことが必須となる。KAGRA においても地下に建設し、地面雑音を下げ、鏡を低温にして熱雑音を減らすなど様々な雑音を減らす工夫がされている。

そこで本修士論文は KAGRA の雑音を低減するという取り組みの中で水による重力勾配雑音に注目したものである。KAGRA は地下に建設されているが、そこでは地下水が発生する。そのため KAGRA ではこの地下水をパイプを通して排出している。このとき、パイプ内の水の質量が変動することで局所的な重力の変動が発生し、それが鏡をゆらす (水との直接的な重力相互作用のゆらぎ) 要因となる。これが水からの重力勾配雑音である。さらにパイプと鏡の距離が近ければこの影響は大きくなり、KAGRA の設計感度に対して雑音となる可能性がある。本修士論文の目的は KAGRA における水の重力勾配雑音を推定し、それが KAGRA に対してどのように影響を与えるかを評価することである。重力勾配雑音は直接測定が難しく、先行研究では理論的に水面を予測することで重力勾配雑音を推定していたが、本論文の試みとして流体シミュレーションによるパイプ内の水の再現を行い重力勾配雑音の大きさを推定した。また、KAGRA の重力勾配雑音を推定することが目的であるため、KAGRA のパイプを再現したシミュレーションを行ったが、そのほかに簡単な真っすぐなパイプを設計しシミュレーションを行った。これは水からの重力勾配雑音が流量やパイプの粗さの違いなどによってどのように変化するのかを確かめるためである。また第三代重力波望遠鏡である Einstein Telescope は地下に建設される予定であり、地下水の重力勾配雑音に影響してくる可能性もあるため、仮定の状況ではあるが ET の設計感度に対しても評価を行った。これらのシミュレーションを行った結果、どの条件においても KAGRA の設計感度より重力勾配雑音の大きさは小さく、KAGRA の感度に影響を与えないという結果が出た。

# 目次

はじめに	i
<b>第 1 章 重力波・重力波望遠鏡</b>	<b>1</b>
1.1 重力波	1
1.2 重力波望遠鏡	3
1.3 重力波望遠鏡における雑音	7
1.4 水の重力勾配雑音	8
<b>第 2 章 KAGRA 内の排水路</b>	<b>11</b>
2.1 KAGRA 内の排水路の配置	11
2.2 KAGRA に使用されている排水パイプの様子	13
<b>第 3 章 水力学</b>	<b>15</b>
3.1 水流の状態の分類	15
3.2 「水路」における流れ	20
3.3 パイプの粗さのパラメータ	23
<b>第 4 章 流体シミュレーションソフトにおける基本設定</b>	<b>26</b>
4.1 流体シミュレーションソフト (Flow-3D)	26
4.2 周辺の基本的環境設定	27
4.3 採用した物理モデル	28
4.4 流体の設定	29
4.5 設計したパイプ	30
4.6 メッシュと境界条件	33
4.7 流体シミュレーションの視覚的な様子	36
4.8 シミュレーションソフトの仕様に起因する回避した設定	37
4.9 Flow-3D からの出力	38
<b>第 5 章 直線状パイプにおける地下水のシミュレーション</b>	<b>41</b>
5.1 シミュレーションの内容	41
5.2 メッシュサイズの比較のシミュレーションの解析結果	44
5.3 初期流量の違いによるシミュレーションの比較の解析結果	47
5.4 相当粗度 (壁面粗さ) の違いによるシミュレーションの比較の解析結果	53
5.5 傾斜の違いによるシミュレーションの比較の解析結果	59

---

5.6	局所的に大きなゆらぎを持つ地点 . . . . .	62
第 6 章	KAGRA のパイプ配置における地下水のシミュレーション	71
6.1	シミュレーションの内容 . . . . .	71
6.2	シミュレーションの解析結果に関する議論 . . . . .	72
第 7 章	重力勾配雑音の解析結果	81
7.1	Python を使用した処理 . . . . .	81
7.2	直線状パイプにおけるシミュレーション . . . . .	84
7.3	KAGRA のパイプ配置 (Y-end のパイプ) における地下水のシミュレーション . . . . .	96
第 8 章	結論	101
8.1	本論文における結論 . . . . .	101
8.2	本研究の将来的な展望 . . . . .	103
付録 A	長方形断面の水路におけるフルード数と跳水の形について	105
付録 B	平均速度公式の定数	106
付録 C	Flow-3D の出力を解析するために作成したプログラム	107
参考文献		135
謝辞		136

# 第 1 章

## 重力波・重力波望遠鏡

### 1.1 重力波

初めに重力波の性質や相対性理論における簡単な説明をする。重力波とは簡単に説明すると空間の歪みが波のように伝わる現象である。まず、重力波は理論的にアインシュタインの一般相対性理論の中で存在が予言されていた。一般相対性理論によると、物体の重力により、周りの空間が歪むとされており、さらに、重力を持つ物体が動くと共に合わせ、空間の歪みが伝搬することも予測されていた。しかし、この重力による空間の歪みはとても小さい変化かつ空間自体の歪みを捉えなければならぬため、高精度な測定を要求される。この重力波を直接検出しようとする試みが長年続けられており、現在ではマイケルソン干渉計を応用した重力波望遠鏡によって検出する方法が主流である。また、重力による空間の歪みは通常とても小さいものなので、大きな歪みを発生させると考えられている強い重力を持つ天体であるブラックホール連星や中性子星連星からの重力波を検出のターゲットとしている。また、重力波は 2015 年にアメリカの重力波望遠鏡により初検出されている。

次に相対性理論の式より重力波の簡単な説明をする。ここでは ([1], P.195~P.201) の重力波の導出を簡単に紹介する。ここでは理想的に平坦なミンコフスキー空間を伝搬する重力波を考える。まず、物質 (の重力) と時空の歪みの関係を表すアインシュタイン方程式は

$$G_{ij} = \frac{8\pi G}{c^4} T_{ij} \quad (1.1)$$

$$G_{ij} = R_{ij} - \frac{1}{2} R g_{ij} \quad (1.2)$$

となっている。ここで  $G_{ij}$  はアインシュタインテンソル、 $G$  は万有引力定数である。次にミンコフスキー計量  $\eta_{ij}$  から微小にずれた空間を考えるとその計量は

$$\eta_{ij} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.3)$$

$$g_{ij} = \eta_{ij} + h_{ij} \quad (1.4)$$

と表せる。 $h_{ij}$  はとても微小なので  $h_{ij}$  の 2 次以降の項は無視できるものとする。そうした場合反変テンソルは

$$\eta^{ij} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.5)$$

$$g^{ij} = \eta^{ij} - h^{ij} \quad (1.6)$$

となる。さらにここで

$$\phi_{ij} \equiv h_{ij} - \frac{1}{2}hg_{ij} = h_{ij} - \frac{1}{2}h\eta_{ij} \quad (1.7)$$

$$h \equiv h^i{}_i = g^{ij}h_{ij} = \eta^{ij}h_{ij} \quad (1.8)$$

を定義する。ここでアインシュタイン方程式に戻る。アインシュタイン方程式の左辺を書き下すと、

$$\square\phi_{ij} - \frac{\partial}{\partial x^i} \left( \frac{\partial\phi_j{}^k}{\partial x^k} \right) - \frac{\partial}{\partial x^j} \left( \frac{\partial\phi_i{}^k}{\partial x^k} \right) + \frac{\partial}{\partial x^k} \left( \frac{\partial\phi_m{}^l}{\partial x^l} \right) \eta^{km} \eta_{ij} = -\frac{16\pi G}{c^4} T_{ij} \quad (1.9)$$

となる。ここで $\square$ はダランベルシアン

$$\square = \eta^{lm} \frac{\partial}{\partial x^i} \frac{\partial}{\partial x^m} \quad (1.10)$$

である。

ここで微小座標変換  $x'^j = x^j + \xi^j$  を用いて式 (1.9) を書き直す。この座標変換による式 (1.7)、式 1.8 で定義された  $h_{ij}$  と  $\phi_{ij}$  の変化をあらわに表すと

$$h'_{ij} = h_{ij} - \eta_{ik} \frac{\partial\xi^k}{\partial x^j} - \eta_{jk} \frac{\partial\xi^k}{\partial x^i} \quad (1.11)$$

$$\phi'_{ij} = \phi_{ij} - \eta_{ik} \frac{\partial\xi^k}{\partial x^j} - \eta_{jk} \frac{\partial\xi^k}{\partial x^i} + \eta_{ij} \frac{\partial\xi^k}{\partial x^k} \quad (1.12)$$

$$(1.13)$$

となる。また  $\phi_i{}^j$  の微分は

$$\frac{\partial\phi_i{}^j}{\partial x'^j} = \frac{\partial\phi_i{}^j}{\partial x^j} - \eta_{ik} \square\xi^k \quad (1.14)$$

となる。さて、ここで微小量  $\xi^k(x)$  について

$$\frac{\partial\phi_i{}^j}{\partial x'^j} = 0 \quad (1.15)$$

となるように取る、という拘束条件を設ける。式 (1.14) よりこの条件は

$$\frac{\partial\phi_i{}^j}{\partial x^j} = \eta_{ik} \square\xi^k \quad (1.16)$$

になる条件とも言える。よって微小量  $\xi^k(x)$  に式 (1.15) の拘束条件をかけ、座標変換される式 (1.9) を整理すると左辺第2、第3、第4項は消え

$$\square\phi_{ij} = -\frac{16\pi G}{c^4} T_{ij} \quad (1.17)$$

と座標変換後のアインシュタイン方程式を書くことが出来る。なお、ここからの式はこの微小座標変換後の式であるためダッシュが付くべきだが便宜上、省くこととする。なお、式 (1.17) の解に対して、 $\square\xi^k = 0$  を満たす座標変換を行う限りその解は座標変換後のアインシュタイン方程式も満たす。これは、この条件は式 (1.16) が0になる条件であり、拘束条件を満たす座標変換となるからである。

ここでこの計算の目的である真空中を伝搬する平面波の重力波について考える。真空である場合式 (1.17) は

$$\square\phi_{ij} = 0 \quad (1.18)$$

となる。このとき、平面波解

$$\phi_{ij} = a_{ij} \exp(ik_l x^l) \quad (1.19)$$

$$k_i k^i = 0 \quad (1.20)$$

を考える。 $k_l$  は4次元の重力波の波数ベクトルである。 $a_{ij}$  は振幅を表すテンソルである。また、式(1.15)の条件より  $a_{ij}k^j = 0$  である。ここで例として重力波が  $x^3$  方向に伝搬しているとする。すると波数ベクトルは

$$k_l = (-k, 0, 0, k), k^l = (k, 0, 0, k) \quad (1.21)$$

となる。さらに、角周波数を  $w = kc$ 、 $x^0 = ct$ 、 $x^3 = z$  とすれば平面波の振動部分は  $\exp(i(-wt + kz))$  となる。

ここで式(1.15)の拘束条件を満たすような微小座標変換

$$x^{r'i} = x^i + \xi^i(x) \quad (1.22)$$

$$\xi^i(x) = \epsilon^i \exp(ik_l x^l) = \epsilon^i / \exp(-ikx^0 + ikx^3) \quad (1.23)$$

を行う。そうすると平面波解の振幅テンソルは

$$a'^i_{ij} = a_{ij} - \epsilon_i k_j - \epsilon_j k_i + \eta_{ij} \epsilon^l k_l \quad (1.24)$$

と変換される。ここで振幅テンソルの表現を分かりやすくするために、 $\epsilon_i$  をトレースレス条件  $a_{ij}\delta^{ij} = 0$  ( $-a_{00} + a_{11} + a_{22} + a_{33} = 0$ ) とトランスバース条件  $a_{ij}k^j = 0$  を満たすように選ぶと

$$\epsilon_0 = -\frac{2a_{00} + a_{11} + a_{22}}{4k} \quad (1.25)$$

$$\epsilon_1 = \frac{a_{01}}{-k} \quad (1.26)$$

$$\epsilon_2 = \frac{a_{02}}{-k} \quad (1.27)$$

$$\epsilon_3 = \frac{2a_{00} - a_{11} - a_{22}}{4k} \quad (1.28)$$

のようにすることができ、これらを整理すると平面波解(重力波)は

$$\phi_{ij} = A^+ e^+_{ij} \exp(-ikx^0 + ikx^3) + A^\times e^\times_{ij} \exp(-ikx^0 + ikx^3) = (h_{ij}) \quad (1.29)$$

$$e^+_{ij} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, e^\times_{ij} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (1.30)$$

と表せる。このように重力波には+と×の2つの振動モードがあり、それぞれ単純な振動モードであることが分かる。なお、この表現はトレースレス条件とトランスバース条件を満たすように選んだので transverse-traceless 表現 (TT 表現) と呼ばれる。

## 1.2 重力波望遠鏡

現在、いくつかの国で重力波を観測するための望遠鏡が建設され運営されている。現在運用されている重力波望遠鏡にはアメリカの LIGO[2]、イタリアの Virgo[3]、ドイツの GEO600[4]、そして日本の KAGRA[5] がある。その中でアメリカにある重力波望遠鏡 LIGO が 2015 年 9 月 14 日に世界で初めて重力波を直接観測した。また、将来的には第三世代重力波望遠鏡として、Einstein telescope[6] や宇宙空間で建設が計画されている LISA[7] のような重力波望遠鏡建設計画が存在する。現在運用されている第二世代重力波望遠鏡はマイケルソン干渉計を利用している。

重力波を検出する仕組みを簡単に説明する。マイケルソン干渉計は図 1.1 のような光学系である。重力波は先述の通り、空間を歪ませる物理現象である。重力波がマイケルソン干渉計を通過すると、マイケルソン干渉計の2つの光路に空間の歪みの影響が現れる。空間の歪みにより、2つの光路の光路長がそれぞれ変化し、マイケルソン干渉計の干渉光の変化として観測することができる。この干渉光を解析することで重力波が通過したことを確認することができる。

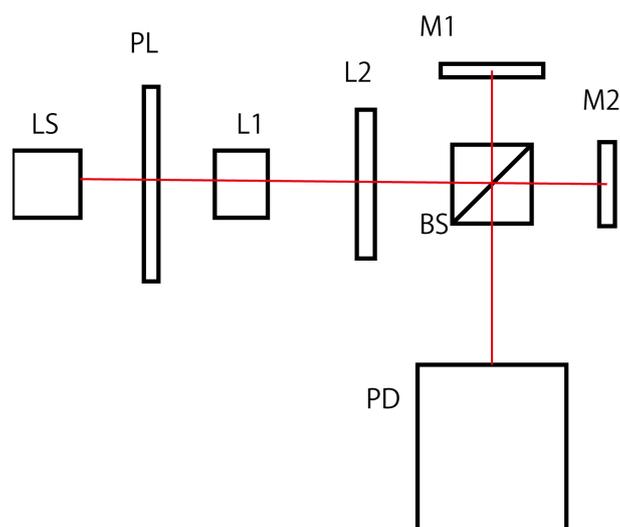


図 1.1: マイケルソン干渉計の簡略図、赤い線は光路を表す。LS はレーザー光源、PL は偏光子、L1L2 はレンズ、BS はビームスプリッター、M1M2 はミラー、PD はフォトディテクターである。

### 1.2.1 大型低温重力波望遠鏡 KAGRA



図 1.2: KAGRA の俯瞰図。KAGRA が L 字型になっている、山の中(地下)に建設されていることが分かる。画像提供：東京大学宇宙線研究所 重力波観測研究施設 [5]

日本の重力波望遠鏡 KAGRA の基本的な情報について紹介する。KAGRA は岐阜県飛騨市神岡町に建設された大型低温重力波望遠鏡である。大型低温重力波望遠鏡と呼ばれるが、「低温」は KAGRA は鏡を低温にしているという特徴を示している。KAGRA もマイケルソン干渉計を利用した重力波望遠鏡である。よってマイケルソン干渉計の形の通り L 字型をしており、腕の長さは 3 km である (図 1.2)。さらに光路長を稼ぐために腕の先端はファブリーペロー共振器となっている。図 1.2 を見ると地下に建設されていることが分かるが、これは地面振動の影響を低減するためである。

KAGRA の設計感度は図 1.3 の通りである。100 Hz 付近の感度が最も良いことが分かる。本論文はこの KAGRA の設計感度を基準にして重力勾配雑音の影響を評価した。なお、O3GK 時の KAGRA の感度は図 1.4 の通りである。KAGRA は建設されてからも感度向上を計っており、[5] によれば図 1.5 のように感度が向上している。

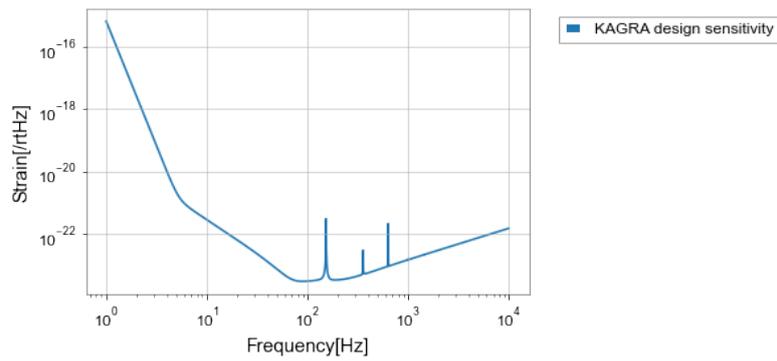


図 1.3: KAGRA の設計感度 [5]。[5] により提供されているデータを元にプロットした。縦軸は  $1/\sqrt{\text{Hz}}$ 、横軸は周波数。

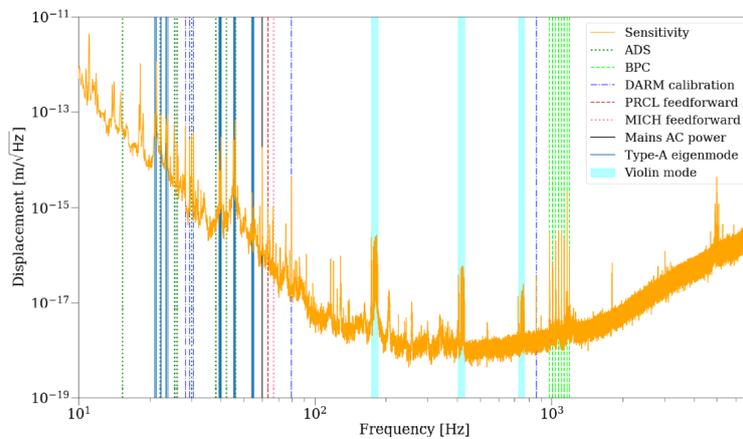


図 1.4: O3GK 時の KAGRA の感度 [10]。オレンジ色が感度である。縦軸は  $1/\sqrt{\text{Hz}}$ 、横軸は周波数。

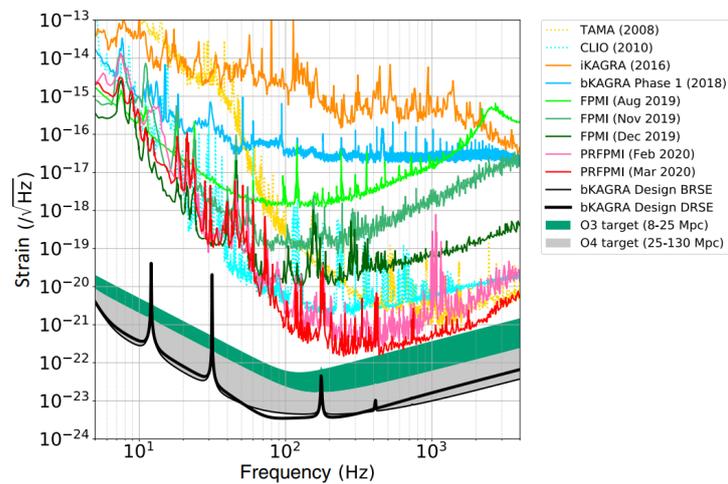


図 1.5: 上グラフ:O3 までの KAGRA の感度向上の記録 [8]。黒色が KAGRA の設計感度であり、感度がそこに近づいていることが分かる。

### 1.3 重力波望遠鏡における雑音

地球に到来する重力波の影響はとて微小であり、マイケルソン干渉計を利用した観測方法においてもとても小さな空間の変化を捉える必要がある。そのため重力波以外の検出器に影響を与える要因は可能な限り取り除く必要がある。ここでは重力波望遠鏡における一般的な雑音について KAGRA を例として説明する。

#### 1.3.1 KAGRA における雑音

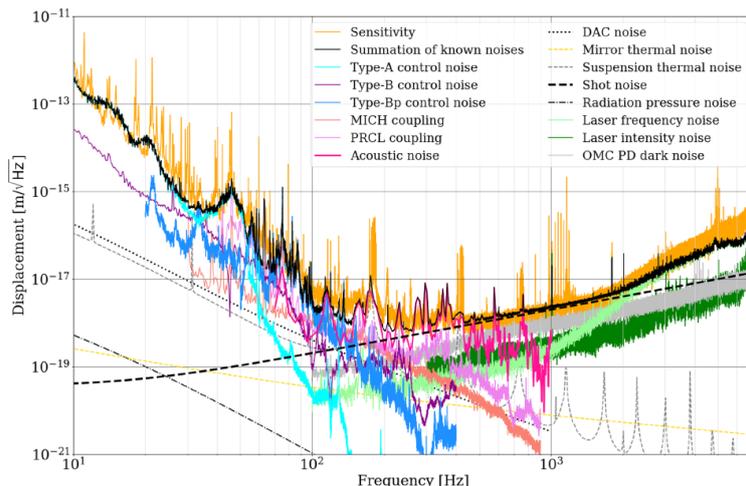


図 1.6: O3GK 時 (2020 年 4 月 12 日) の KAGRA の雑音。測定された雑音について KAGRA の感度と載せている。[10]

KAGRA では様々な雑音についての推定やモニタリングが行われている (図 1.6)。雑音は大きく、地面や機器からの振動によるもの、熱雑音、光源 (レーザー) 由来の雑音、その他 (周辺環境の状況などによる雑音) がある [9]。ここでは KAGRA における雑音として O3GK 時 [10] に言及されている雑音について紹介する。基本的には低周波領域では鏡に伝わる地面振動や重力勾配雑音などの影響が大きく、高周波領域ではショットノイズや量子圧雑音などの影響が大きくなる。KAGRA がターゲットにしている 100 Hz 付近では 100 Hz~400 Hz で周辺の機器から発生する音響雑音や鏡の懸架に使用しているサファイアファイバーのバイオリンモード (横波の定在波) や熱雑音が影響大きいとされている。KAGRA では上記を含めた各雑音源の排除や減衰などを行い設計感度に到達できるよう対策を行っている。次節では本研究で取り扱う重力勾配雑音について説明する。

#### 1.3.2 重力勾配雑音 (Newtonian noise)

重力勾配とは、重力 (万有引力) の勾配を指す言葉である。重力波望遠鏡における重力勾配雑音は、検出器に使われる鏡が、周辺の物体が動くことで生じる局所的な重力の変化の影響を受けることで発生する。現在、重力波望遠鏡における重力勾配雑音の要因として、地面の変動、空気の密度の変動、水の変動、その他周辺機器の振動などが考えられている。KAGRA においても本研究である水以外にも機器の揺れによる重力勾配雑音の推定 [11] が行われている。それぞれの検出器において環境の違いなどで重力勾配雑音の影響の程度や注目すべき雑音源の違いはあるが、第三世代重力波望遠鏡の感度は重力勾配雑音に影響される可能性が高く、推定や低減する方法の研究が進められている。

## 1.4 水の重力勾配雑音

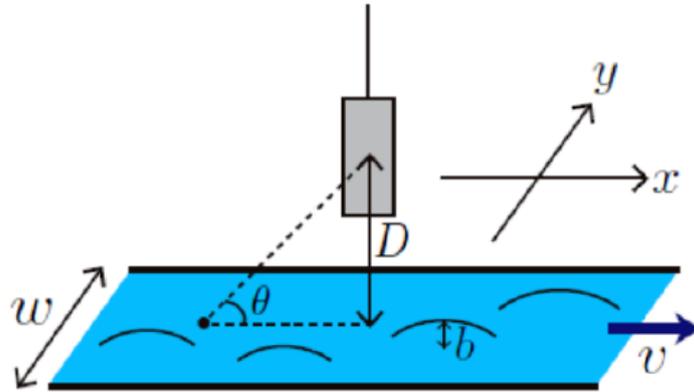


図 1.7: 水の重力勾配雑音の模式図 [12]

水 (水面) が変動することによる質量の移動が万有引力の揺らぎとなり、鏡 (テストマス) に対して重力勾配雑音となる。本研究では水の変動によって生じる重力勾配雑音を推定することを目的に流体シミュレーションを行った。また、KAGRA 内では壁面、壁中にも地下水が伝わっていると推定され、その水の質量変化によっても重力の揺らぎが発生すると考えられる。本論文においては鏡に近い位置を通るパイプ内の水から発生する重力勾配雑音のみを考える (図 1.7)。KAGRA 内のパイプの情報については後述する。

### 1.4.1 先行研究による水の重力勾配雑音の見積り

先行研究 [13]、[14] では 2 種類の水面の動きのモデルが提案されている (図 1.8)。モデル 1 は水面が各地点において変動し、空間方向の伝搬については考えないモデルである。モデル 2 は水面で出来た波がそのまま形を保ったまま流れるというモデルである。それぞれのモデルにおいて重力勾配雑音が見積もられており、図 1.8 のようになっている。この見積りによれば水面の動きがモデル 1 だった場合に KAGRA の設計感度を上回ることになり、重力勾配雑音が KAGRA にとって問題になる。しかし、これらの水面の動きのモデルは実際の水の動きを参照しておらず、現実には水がどのような動きで流れているか不明だった。そこで、本研究では実際の水の動きをシミュレーションを用いて再現し、重力勾配雑音を推定することを目的とする。

### 1.4.2 定式化

先行研究 [12] を参考にするとテストマス (質点) と水の重力勾配雑音の式は以下の通りである。重力勾配雑音は重力のゆらぎによるものより、雑音の定式は水とテストマスの重力相互作用の足し合わせの式で表せる。よってテストマスが水の質量変化より受ける光軸方向への加速度は

$$a = \iint \frac{G\rho b(x, y, t) \cos \theta}{(x - x_{TM})^2 + (y - y_{TM})^2 + (z - z_{TM})^2} dx dy \quad (1.31)$$

$$= G\rho \iint \frac{b(x, y, t) \cdot (x - x_{TM})}{((x - x_{TM})^2 + (y - y_{TM})^2 + (z - z_{TM})^2)^{3/2}} dx dy \quad (1.32)$$

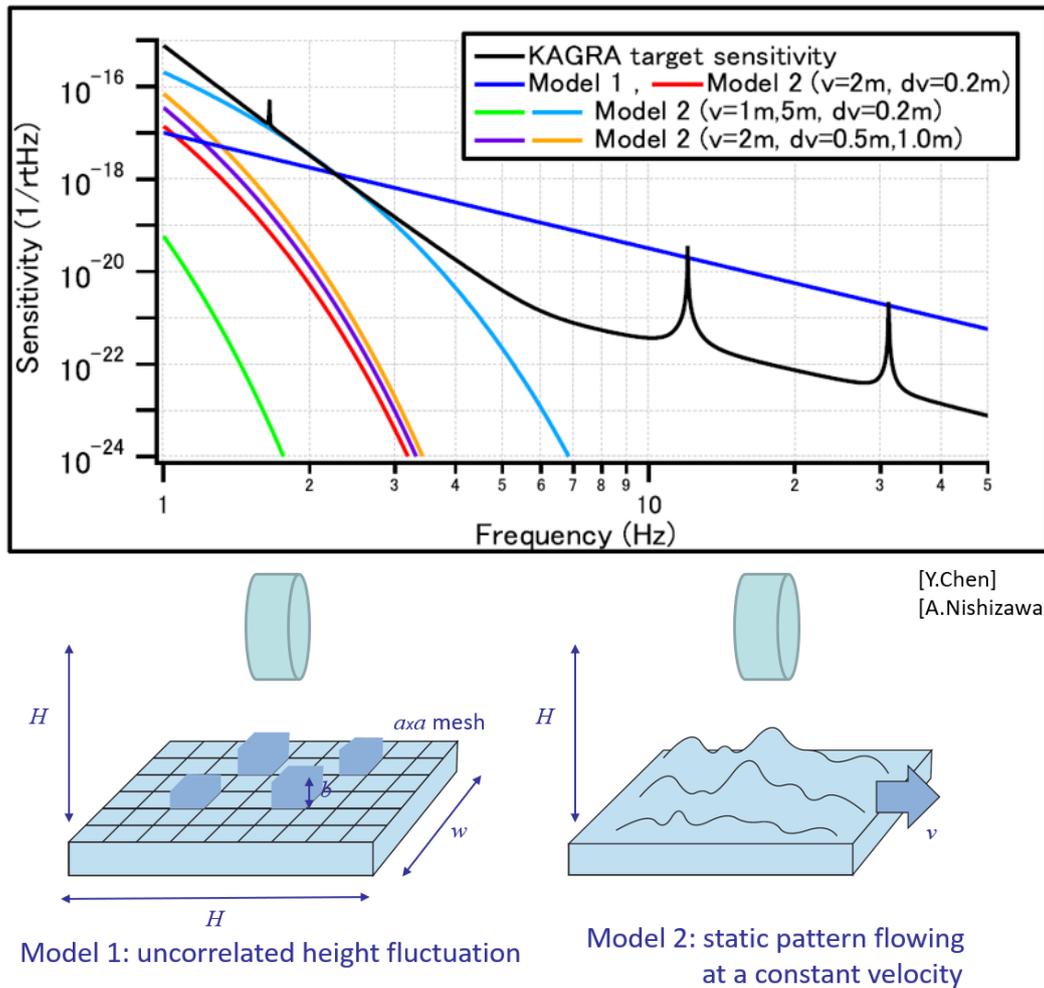


図 1.8: [13]、上グラフはモデル 1、モデル 2 それぞれの水面からの重力勾配雑音の見積。モデル 2 では重力勾配雑音は高周波側で急激に小さくなっており、影響は限定的だが、モデル 2 の場合重力勾配雑音は高周波側での減少が小さく、影響が広い周波数で現れることが予測される。下図はモデル 1 とモデル 2 のイメージ図である。モデル 1 では水面がそれぞれの地点で変動し、モデル 2 では水が形を保って移動する様子を表している。

となる。ここでテストマスは  $(x_{TM}, y_{TM}, z_{TM})$  の位置にあり、 $(x, y, z)$  は対象とする水の素片のそれぞれの重心座標である。パラメータ  $b$  はある  $x, y$  座標における水深であり、時間変化する。テストマスの具体的な座標については後述する。

また、加速度を周波数空間で見ることがあるので  $a$  のフーリエ変換を  $\tilde{a}(\Omega)$ 、パワースペクトルを  $S_a$  とし、ひずみ感度の次元に直したものを  $S_h$  とすれば

$$\langle \tilde{a}(\Omega)\tilde{a}(\Omega') \rangle = S_a \delta(\Omega - \Omega') \tag{1.33}$$

$$\sqrt{S_h} = \frac{1}{L\Omega^2} \sqrt{S_a} \tag{1.34}$$

となり雑音を KAGRA の感度と比較することができる。ここで  $L$  は KAGRA の腕の長さ 3000 m であり、 $\Omega$  は角周波数である。

### 1.4.3 先行研究を踏まえた本研究の意義

先行研究では理論的にパイプ内の水面の様子を考察し、重力勾配雑音を推定していた。しかし先行研究で行われていた理論的な水面の様子考察では、対象が流体であることに加えて現実の諸条件に水面が影響される KAGRA のパイプの水面を予測することには限界があった。そこで本研究ではパイプ内の水面をシミュレーションにより再現することでより正確な現実の重力勾配雑音の推定を目的とした。さらに、シミュレーションの結果だけでなく先行研究にある水面の仮説や重力勾配雑音の式なども参考にして、KAGRA における水からの重力勾配雑音について正確な推定を行う。またシミュレーションを行った理由として、理論では推定に限界があることの他に KAGRA のパイプは透明ではなく中の様子が観察できないことや水面自体の揺れを高精度で測定することが難しいことがあげられる。

本研究は宗宮研究室の先輩方により、いくつかのシミュレーションや重力勾配雑音の推定が行われている。本論文では、使うシミュレーションソフトや重力勾配雑音の計算方法(本論文の解析に使用したプログラムは新たに作成した)が異なるため直接シミュレーションや解析を参考にしないが、本論文では先輩方の学位論文より多くの知見を得て執筆している。

## 第 2 章

# KAGRA 内の排水路

### 2.1 KAGRA 内の排水路の配置

KAGRA は先述の通り、地下に建設されている。そのため、KAGRA 内部には地下水が染み出しており、図 2.2 が実際の KAGRA 内部の様子である。その地下水を排出するためのパイプが設置されている (図 2.1)。基本的には KAGRA の腕 (レーザー) の方向に沿って配置されている。本研究ではパイプと鏡の位置が比較的近い位置にある

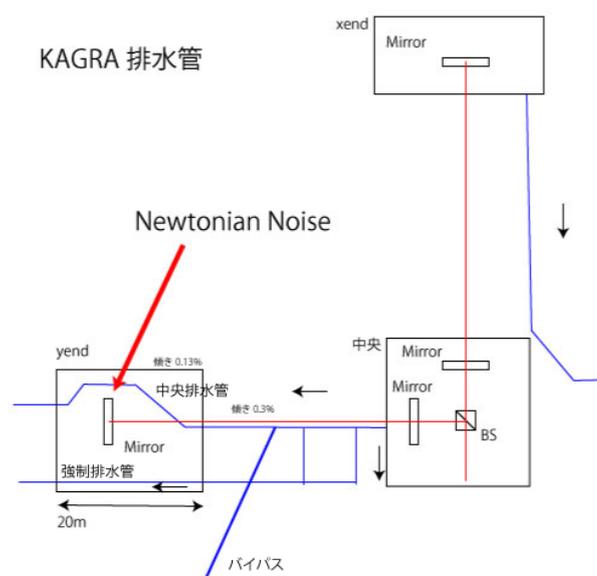


図 2.1: 現在の KAGRA の排水路の配置。KAGRA を上から見たもの。青い線が排水路の位置を表している。壁や地面から染み出した水は青い部分の排水路を通り KAGRA 外へ排出される。水流の方向は黒矢印で表してある。実際には縦穴を通していている部分があるが、Y-end より離れた場所であるためここでは表示を省略している。

Y-end の鏡が水の重力勾配雑音の影響受けやすいと考えている。そのため、本論文におけるシミュレーションは Y-end の状況を想定したものとなっている。Y-end に向かうパイプは Y-end 直前までは傾斜 0.3 %、Y-end 内では 0.3 % となっている。本論文におけるシミュレーションでは直線状のパイプと Y-end 内のパイプを再現したパイプの 2 種類のパイプを使用するが、前者は傾斜 0.3 %、後者は 0.13 %を採用している。



図 2.2: KAGRA 内部で地下水が染み出す様子

### 2.1.1 Y-end における排水パイプの配置

本論文において重力勾配雑音の影響を受ける可能性があるとして想定している Y-end の鏡と、雑音の発生源となる Y-end 内の排水パイプの位置関係について説明する。

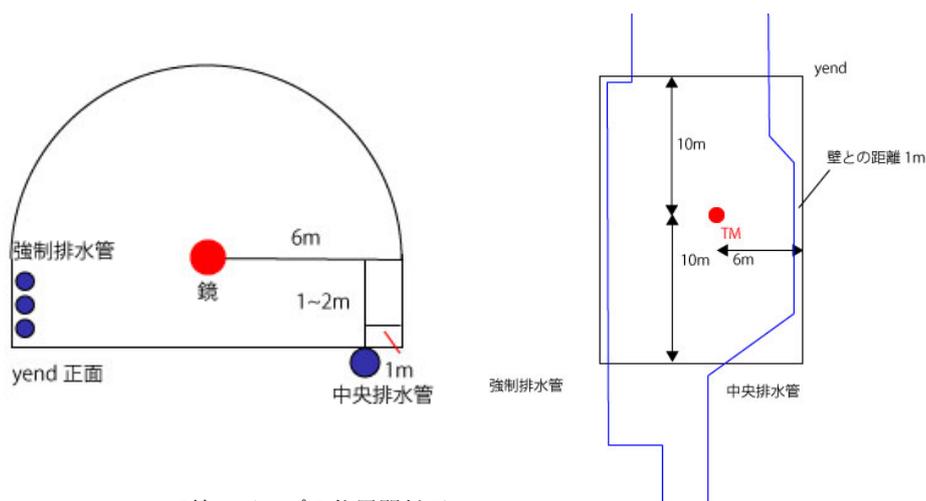


図 2.3: Y-end において鏡とパイプの位置関係図。壁際の 3 本は強制排水管、地中にあるのが中央排水管。水の流れは手前から奥に流れている。赤色が鏡である。

図 2.4: Y-end を上から見た時の配管の配置。水の流れは下から上である。赤色が鏡である。

Y-end 内の排水パイプの配置は図 2.3、図 2.4 のようになっている。Y-end 内には合計 4 本の排水パイプが通っており、そのうち 3 本は強制排水管と呼ばれ壁面に沿って通っており、1 本は中央排水管と呼ばれ地中を通っている。本論文では鏡に近い中央排水管内の水が鏡に与える重力勾配雑音を推定する\*1。中央排水管の鏡と一番近い位置は鏡とは横 5m、縦 1m 2m の位置にある。強制排水管は図 2.1 の通り yend に向かう途中のいくつかの地点で中央排水管から分離され、中央排水管の水量が増えすぎないように調節する役目を果たしている。図 2.5 は Y-end に向かう強制排水管の写真である。Y アームに沿って設置されている。

\*1 中央排水管は地中に埋まっているという事情からパイプを移設する工事が大掛かりなものになることが予想される。水の重力勾配雑音の影響が大きい場合、パイプを移設する以外の雑音低減の方法を模索したい。



図 2.5: Y-end に向かう強制排水管の様子。Y アームに沿ってパイプが設置されている。



図 2.6: Y-end 内の様子。この部屋の床下に中央排水管が通っている。中央に鏡が設置されている。

## 2.2 KAGRA に使用されている排水パイプの様子

Y-end で使用されている排水パイプの自体の様子について説明する。図 2.7 は KAGRA 敷地内にあるパイプの写真である。このパイプは株式会社クリモトポリマーにより提供されている [15]。さらに、写真のパイプは KAGRA 内部で使用されているものと同一である。KAGRA 内部で使用されているものは肉眼で観察できる場所にあるものが少なく、本節では敷地内に放置されているものを観察している。このパイプは外側が蛇腹形状になっており、内側も蛇腹形状による多少の凹凸が認められる。しかし内側の凹凸は微小なため、本論文で設計したパイプには反映されていない。さらに内径は実際に測り 0.4m であることを確認できた。

### 2.2.1 実際の水流の観察

KAGRA のパイプは透明ではないため、中の水を肉眼で観察することはできない。さらに、中央排水管は Y-end の地下に埋まっているため肉眼でみることはできない (図 2.6)。ここではパイプの出口や Y-end の先にある肉眼で観察



図 2.7: KAGRA 敷地内で放置されているパイプの写真。このパイプと同様のパイプが KAGRA 内に設置され、地下水を排出している。内径は左写真にあるように 0.4m である。パイプ外部は蛇腹形状であり凹凸があるが、パイプ内部では外部ほど凹凸はない。

できる水路の様子について述べる。図 2.8 は強制排水管の出口の様子であり、Y-end 先の水路に水が流れ込む様子である。この写真の撮影時 (2021 年 8 月) では強制排水管の出口では半分より少し低いところまで水があることが分かる。



図 2.8: Y-end の先のパイプから排出される水の様子 (2021 年 8 月)。このパイプは強制排水管の出口であり Y-end の先にある水路に水を流している。パイプの出口であればこのように観察できる場所が存在する。

## 第3章

# 水力学

本研究は水の変動によって発生する重力勾配雑音を推定することが目的である。そのためにはパイプ内で水自体がどのような状態であるか、どのように変化するのかといったことを知る事が重要である。推定自体には流体シミュレーションを用いるが、流体力学や水力学といった分野の視点からシミュレーション結果を考察することにより、重力勾配雑音とパイプの条件の違いによる水の状態変化の関係について知ることができる。また流体シミュレーションのパラメータとして必要な物理量も存在するためここで紹介する。

### 3.1 水流の状態の分類

水の表面の変動が直接万有引力の変化となるため、水の流れの状態を区別することが重要となる。ここでは2種類の、流体を区別する概念を紹介する。それぞれ水の重力勾配雑音を推定するために重要である水面の水の変動(質量変化)の振る舞いを理解する上で欠かせないものである。

#### 3.1.1 層流と乱流

流れの状態の区別として層流と乱流という区別がある。流体力学の分野では層流と乱流の場合で分けて理論が考えられている場合があり、流れを物理的に理解する上で有効な区別となっている。層流と乱流の区別は1883年にレイノルズ(Osborne Reynolds)が行った実験により発見された[16]。この実験では細長いガラス管に水を流し、そこに色素を注入することで流れの特徴を観察している。その結果、流れには2つの種類があることが判明した。1つは、水の速度が遅い時に、色素の線が直線になりはっきりと観察できる場合である。このような状態を層流(laminar flow)と名付けた。もう1つの場合では水の流れが速い時に、色素が拡散し、不規則に動いている様子を観察できた。このような状態を乱流(turbulent flow)と呼ぶ。乱流の場合では水流の時間平均が一定であっても各水粒子が常に不規則に動くことから速度変動が一定になることがないことが特徴である。

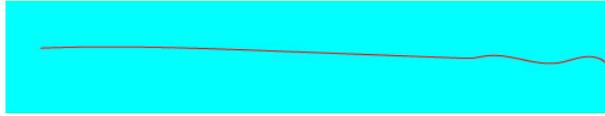
#### レイノルズ数

レイノルズは層流と乱流を区別する指標としてレイノルズ数を見出した。管内のレイノルズ数は

$$Re = \frac{Vd\rho}{\mu} = \frac{Vd}{\nu} \quad (3.1)$$

とされる[16]。ここで $d$ は管の内径、 $V$ は流体の平均速度、 $\rho$ は流体の密度、 $\mu$ は流体の粘度である。あるレイノルズ数 $Re_c$ より小さいレイノルズ数を持つ場合を層流、大きい場合を乱流と見なし、このレイノルズ数を臨界レイノルズ数と呼ぶ。臨界レイノルズ数は流体が流れる条件によって変化し、また流体の速度を下げたとき乱流から層流とする時

(a) 層流



(b) 乱流



図 3.1: レイノルズの実験。(a) が層流の水流に赤インクを注入した際の結果の略図である。このときインクは広がらずに下流まで流れる。(b) は乱流の水流にインクを流した状態である。乱流の場合インクは下流方向に流れる際乱れを伴って流れる。

のレイノルズ数の値と、速度を上げていき層流から乱流に変化させるときの値が違うなど、様々な条件によって変化する。一般的な流れでは臨界レイノルズ数は 2300 程度であると知られている。ただし、式 (3.1) は管内が満水状態の計算式であり、後述する水路における式と少し異なっている。

### 3.1.2 常流、射流、(跳水)

水力学の分野における水流の分類である、常流、射流について説明する。跳水については次節で説明する。これらの現象は後述する水路 (開水路) においてのみ発生する。この節では [16]、[17]、[18]、を参考に記述した。

#### 常流と射流 (流速との比較)

[17](p.115、p.116) に記載される常流と射流の定義について紹介する。ある水面を攪乱すると水面に波が立つ。この波を”小重力波”<sup>\*1</sup>と呼ぶ。この波の伝搬速度は  $\sqrt{gD}$  で示される。 $g$  は重力加速度、 $D$  は水理水深 (流積を水路幅で割ったもの、後述する径深と同様) である。ここで、流速  $V$  のある開水路を考える。この開水路で伝搬速度  $\sqrt{gD}$  の小重力波を発生させた時に

$$V < \sqrt{gD} \quad (3.2)$$

となる状態を常流 (Subcritical flow) と呼ぶ。常流では水面で発生した波の伝搬速度が流速を上回っており、水面で発生した波が下流だけでなく上流側にも伝わることを示している。反対に

$$V > \sqrt{gD} \quad (3.3)$$

<sup>\*1</sup> 宇宙で発生する”重力波”とは言葉は同一だが全く違うものである。英語名では”gravity wave”(流体力学などで扱われる流体による波)、“gravitational wave”(宇宙で発生する空間を伝わる歪みによる波)と区別される。

となる状態を射流 (Supercritical flow) と呼ぶ。つまり、射流の状態では水面で発生した波の伝搬速度を流速が上回っており、水面で発生した波は上流側には伝わらず下流のみに伝わることを示している。さらに

$$V = \sqrt{gD} \quad (3.4)$$

となる時の流れを限界流 (critical flow) と呼び、この時の水深を限界水深と呼ぶ。限界水深については詳しく後述する。また常流、射流の区別をする場合

$$Fr = \frac{V}{\sqrt{gD}} \quad (3.5)$$

で定義されるフルード数  $Fr$  を用いてこの値が1となる時を基準に判断する。これらの関係は、流速  $V$  と水理水深  $D$  によって決まるので射流と常流を区別するにはこの2つの値が重要となる。また常流と射流の波の伝搬の特徴から射流よりも常流のほうが上流側と下流側両方向に波が伝わるため、広い範囲で波によって水面が揺れる。ただし、その場合でもモデル2のように形を保っていた場合重力勾配雑音に与える影響は小さいと考えられる。次に常流と射流を流体力学の式から見る。

#### ベルヌーイの定理

常流、射流からは外れるが、常流と射流の状態を流体力学の観点から説明するためにまず流体力学の基礎であるベルヌーイの定理について ([16]、P.42、P.43) を元に紹介する。粘性のない流体の定常流の運動方程式を考える。流体の密度を  $\rho$  とし、断面積  $A$  流れ方向の長さ  $ds$  の微小部分  $A ds$  を考える。このとき、微小部分の運動方程式は流速を  $V$  とし

$$\rho A ds V \frac{dV}{ds} = -A \frac{dp}{ds} ds - \rho g A \frac{dz}{ds} ds \quad (3.6)$$

$$V \frac{dV}{ds} = -\frac{1}{\rho} \frac{dp}{ds} - g \frac{dz}{ds} \quad (3.7)$$

となる。これはオイラーの運動方程式と呼ばれる。非圧縮性流体とすると密度が一定となり、さらにこの式を  $s$  に沿って積分すると

$$\frac{V^2}{2} + \int \frac{dp}{\rho} + gz = \text{定数} \quad (3.8)$$

$$\frac{V^2}{2g} + \int \frac{dp}{\rho g} + z = \text{定数} \quad (3.9)$$

となる。これをベルヌーイの式という式 (3.9) は長さの次元を持っており、それぞれ左辺第1項を速度ヘッド、第2項を圧力ヘッド、第3項を位置ヘッドと呼び、右辺の定数を全ヘッドと呼ぶ。これはある流れ方向の線に屋いしては全ヘッドが一定となることを示している。また、式 (3.8) はエネルギーの次元を持つので、流体の運動エネルギー、圧力、位置エネルギーが保存していることも示している。

#### 限界水深、及び常流と射流の状態の導出

フルード数が1になる時の水深、限界水深を [16] を参考に導出する。水路が傾いているとき、水路の底面を基準とした全ヘッドを  $H_o$  としベルヌーイの式は

$$H_o = \frac{v^2}{2g} + y + \frac{p}{\rho g} \quad (3.10)$$

$$= \frac{v^2}{2g} + h \quad (3.11)$$

となる [16]。  $y$  は水底からの高さ、  $p$  は圧力、  $\rho$  は密度、  $h$  は水深である。さらに水路の断面積を  $A$ 、流量を  $Q$  とすると

$$H_o = \frac{1}{2g} \left( \frac{Q}{A} \right)^2 + h \quad (3.12)$$

となる。ここで断面が幅  $b$  の長方形の水路を考えると  $q = \text{流量}/\text{水路幅}$  として、  $Q = bq$ 、  $A = bh$  とすることができ、

$$V = \frac{Q}{A} = \frac{q}{h} \quad (3.13)$$

$$H_o = \frac{1}{2g} \left( \frac{q}{h} \right)^2 + h \quad (3.14)$$

となる ( $V$  は流速)。  $H_o$  が最小になるときの水深の値は、式 (3.14) を微分してその値が 0 になるときが  $H_o$  の最小値となるので、

$$\frac{dH_o}{dh} = -\frac{q^2}{gh^3} + 1 = 0 \quad (3.15)$$

$$h = \left( \frac{q^2}{g} \right)^{1/3} \quad (3.16)$$

となり、最小の  $H_o$  を与える水深  $h$  の値式 (3.16) が分かる。さらにこの値を式 (3.14) に代入すると最小の  $H_o$

$$H_{omin} = \frac{3}{2} \left( \frac{q^2}{g} \right)^{1/3} \quad (3.17)$$

が得られる。「単位幅流量 ( $q$ ) が一定のとき、水路底面を基準とした全ヘッド  $H_o$  が最小になる水深」を限界水深  $h_c$  式 (3.16) と呼ぶ。式 (3.16) と式 (3.17) を  $h_c$  で書き表すと

$$h_c = \left( \frac{q^2}{g} \right)^{1/3}, H_{omin} = \frac{3}{2} h_c \quad (3.18)$$

となる [16]。さらにこの時の流速  $V_c$  を考える。流速は単位幅流量  $q$  を水深  $h_c$  で割ればよいので

$$V_c^2 = \frac{q^2}{h_c^2} = g \frac{h_c^3}{h_c^2} = gh_c \quad (3.19)$$

$$V_c = \sqrt{gh_c} \quad (3.20)$$

となる。これは式 (3.4) と同一であり、限界流では流速と小重力波の伝搬速度が等しいことが分かる。つまり限界流とは全ヘッドが最小となる状態で、水深と全ヘッドが式 (3.18) のようになる状態である。

ここで式 (3.2)、式 (3.3) を考える。単位幅流量  $q$  が一定のとき式 (3.13) より、流速  $V$  が遅ければ水深  $h$  は増加し、流速  $V$  が速ければ水深  $h$  は浅くなる。つまり、限界流を基準に考えた時、  $V < V_c$  かつ  $h_c < h$  となる状態では

$$V < V_c = \sqrt{gh_c} < \sqrt{gh} \quad (3.21)$$

となる。これは、式 (3.2) と同一であり、流速と小重力波の伝搬速度の関係から常流となる。つまりこれを満たす条件より流速と小重力波の伝搬速度の関係だけでなく、常流は限界水深より深い水深で限界流より流れが遅い状態とすることができる。反対に  $V > V_c$  かつ  $h_c > h$  となる状態では

$$V > V_c = \sqrt{gh_c} > \sqrt{gh} \quad (3.22)$$

となる。これは式 (3.3) と同一になり、流速と小重力波の伝搬速度の関係から射流となる。

### 常流と射流まとめ

先述した常流と射流に関することをまとめると常流と射流と限界流は以下のようになる。

$$\text{常流} : h > \left(\frac{q^2}{g}\right)^{1/3}, h > \frac{2}{3}H_o, \frac{V}{\sqrt{gh}} < 1 \quad (3.23)$$

$$\text{射流} : h < \left(\frac{q^2}{g}\right)^{1/3}, h < \frac{2}{3}H_o, \frac{V}{\sqrt{gh}} > 1 \quad (3.24)$$

$$\text{限界流} : h = \left(\frac{q^2}{g}\right)^{1/3}, h = \frac{2}{3}H_o, \frac{V}{\sqrt{gh}} = 1 \quad (3.25)$$

ここまでをまとめると、常流と射流を小重力波の伝搬速度と流速で比較する見方と、全ヘッドが最小になる状態を基準として見分ける見方の2種類があることがわかる。

### 3.1.3 跳水

跳水とは水流の状態が”射流から常流”に変化する時に遷移領域(限界流領域)で発生する状態である。跳水では水位が急激に上昇し、水の変化も激しくなる。よって水面も激しく揺れるため、定常流(後述)の場合で水からの重力勾配雑音を考えた時に最も雑音源となる現象と言える。跳水は限界流領域で発生するが変化が激しく、限界流における議論とは別に考える必要がある。ここでは跳水についての基本的な式について [17] と [18] を参考に紹介する。

#### ベランジャーの式

跳水部分では水位が大きく変化する。長方形断面水路において、跳水の水位は跳水の水平方向の運動方程式

$$\rho q B V_2 - \rho q B V_1 = w \frac{1}{2} B h_1^2 - w \frac{1}{2} B h_2^2 \quad (3.26)$$

を整理することで得られるベランジャーの式

$$\frac{h_2}{h_1} = \frac{1}{2} \left( \sqrt{8Fr_1^2 + 1} - 1 \right) \quad (3.27)$$

$$\frac{h_1}{h_2} = \frac{1}{2} \left( \sqrt{8Fr_2^2 + 1} - 1 \right) \quad (3.28)$$

で与えられる ([17], P.124)。 $\rho$  は密度、 $q$  は単位幅流量、 $B$  は水路幅、 $V_1$  は開始地点の  $V_2$  は終了地点の流速、 $w$  は単位重量 ( $\rho g$ )、 $h_1$  は跳水開始地点の水深、 $h_2$  は跳水終了地点の水深、 $Fr_1$  は跳水開始地点のフルード数 ( $Fr_1 = V_1/\sqrt{gh_1}$ )、 $Fr_2$  は跳水開始地点のフルード数 ( $Fr_2 = V_2/\sqrt{gh_2}$ ) である。このように跳水においてどの程度水位が上昇するか理論的に予測することができる。

#### 跳水によるエネルギー損失

([17], P.124, P.125) による計算を紹介する。跳水の開始地点と終了地点でベルヌーイの定理を考えると

$$H_L = H_1 - H_2 = \left( \alpha_1 \frac{V_1^2}{2g} + h_1 \right) - \left( \alpha_2 \frac{V_2^2}{2g} + h_2 \right) \quad (3.29)$$

となる。 $H_L$  は跳水による損失ヘッド、 $H_1$  は開始地点の  $H_2$  は終了地点の全ヘッドである。長方形断面の水路の場合  $\alpha = 1$  となり、式 (3.26) から

$$H_L = \frac{V_1^2}{2gh_1} h_1 \left( 1 - \left( \frac{h_1}{h_2} \right)^2 \right) + h_1 - h_2 \quad (3.30)$$

$$= \frac{(h_2 - h_1)^3}{4h_1 h_2} \quad (3.31)$$

となる [17]。また相対エネルギー損失は

$$\frac{H_L}{H_1} = 1 - \frac{Fr_1^2 (h_1/h_2)^2 + 2(h_2/h_1)}{Fr_1^2 + 2} \quad (3.32)$$

となる [17]。例えば  $Fr_1$  を 5~9 としたとき相対エネルギー損失は 50 %~70 % となる。以上の式から跳水が発生した時のエネルギー損失を知ることが聞出る。

### 跳水の形

跳水の長さに関しては理論的に求めることは難しい。いくつかの形の水路や定常流における跳水の長さに関しては実験から式が見積もられているが、本研究においては対象にするパイプに適用することは難しい。参考として付録 A にて、[18] に記載される長方形断面の水路におけるフルード数と跳水の形の関係について紹介する。跳水の位置に関しても特定することは真っすぐな理想的な水路においては理論的に計算する方法があるが、KAGRA の Y-end におけるパイプのように途中で曲がっているなどして複雑な形をしている場合そのあらかじめ予測することは難しい。よって本論文で行ったようにシミュレーションによりその様子を観察する方法が跳水の発生や様子を予測する上で簡単な方法であると言える。

### KAGRA の水路における跳水

KAGRA の水路でも実際に跳水が起こっている様子が観察できた (図 3.2)。このように肉眼で観察しても白波が立ち水面に大きく影響を与える現象であることが分かる。よって跳水がテストマスに近い水域で起こる場合、重力勾配雑音も大きくなることが予想される。

## 3.2 「水路」における流れ

本論文で行った実験ではパイプ内の水流を想定しているが、実際の水量と照らし合わせて実験を行うと、パイプ内が満水になっていない状態がほとんどであることが判明している。このように川や水路と同じ様に自由表面を持つ場合、「水路」または開渠 (channel)、開水路 (open channel) と呼ばれる [16]。本実験の環境ではパイプ内の水の流れは「水路」である。

パイプ内に水のある量以上流せば、自明にパイプ内は満水になり、水路とは呼べない状態になる。この場合は水の質量変化がなく重力勾配雑音は発生しないと考えられる\*2。そのためパイプが満水の場合における水の状態に関する流体力学的な研究は数多くされているが、本論文では紹介しない。

\*2 水自体の密度が局所的に急激に変化することがあれば発生する可能性はあるが、KAGRA 内のパイプでそのような現象が起こることは予測されない。



図 3.2: KAGRA 内の水路における跳水の観察。Y-end の奥にある水路、Y アームに沿ったパイプを流れている水の下流である。この水路の断面は直方体になっている。水路中央にある白波が跳水と考えられる。この写真では左側 (上流) が射流になっており、右側 (下流) が常流になっていると考えられる。この写真では判別しにくいですが、跳水手前までの射流領域は水深が浅く、跳水部分で水位が上がり、常流部分では水深が深くなっていることが分かる。これは射流、常流の定義にも矛盾しない。

### 3.2.1 水路における流れの区分

前節では水面の様子に着目して 2 種類の流れの区分を紹介した。水路には他にも流れの分類がある (図 3.3)。水路における流れは大きく 2 つに区別される。水路の流量に時間的な変化がない流れを定常流と呼び、それ以外の流れを非定常流と呼ぶ。現実的に厳密に流量が完全に一定の流れは少ないが、実用的に、短時間の平均流量がほぼ一定であれば定常流とすることもある。さらに定常流の中で空間的な変化 (水深の変化) がない流れを等流と呼び、それ以外の流れを不等流と呼ぶ。定常流の等流は、断面形や壁面の状態および勾配が一定の長い直線水路で起こる [16]。本論文における実験の水路の流れは、一定の条件で十分に長い時間を置いたシミュレーションであるため、定常流だと考える。定常流とした上で前節で紹介した、層流、乱流、常流、射流、跳水を考える。

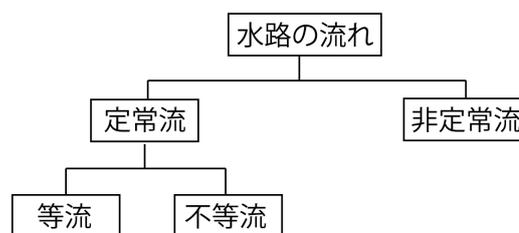


図 3.3: 水路における流れの区分。まず、流れは流量に時間的な変化がない定常流とそれ以外の非定常流に分けられる。本論文で行った研究は一定の条件で十分に時間を置いた後の水流である定常流であると考えられる。さらに、定常流の中で空間的な変化がない流れを等流と呼び、それ以外の流れを不等流と呼ぶ。また、先述した跳水は定常流の不等流にあたる。常流、射流が等流にあたるか不等流であるかは状況によって異なる。

### 3.2.2 水路のレイノルズ数

先述したレイノルズ数の式 (3.1) は満水の管内における式であった。本論文における実験に適合する水路におけるレイノルズ数の計算式は

$$Re = \frac{mV\rho}{\mu} \quad (3.33)$$

となる。ここで  $m$  は径深 (水力平均深さ)、 $V$  は流体の平均速度、 $\rho$  は流体の密度、 $\mu$  は流体の粘度である。径深とは水路の断面を見た時に流体の面積を流体が接する縁の長さで割ったものである。

さらに本研究で再現するパイプの径深は (図 3.4) のようになる。

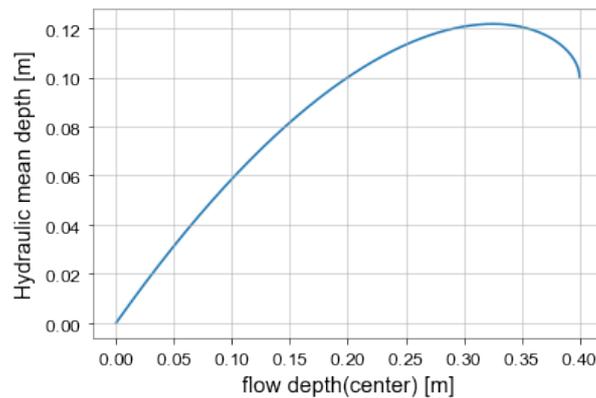


図 3.4: 縦軸は径深、横軸はパイプの中心の水深 (パイプの底と水面の直線距離) である。KAGRA で使用している内径 0.4m の円柱型のパイプを参考に計算した。水深がパイプの半分 (水深 0.2m) の場合と満水の時 (水深 0.4m) 径深は 0.1m である。水深 0.3m 付近で径深は最大となる。

水路のレイノルズ数では  $Re < 500$  で層流、 $Re > 2000$  で乱流となる。本論文で行った実験の条件でレイノルズ数を式 (3.33) より見積もると

$$m \approx 0.04 \sim 0.12m \quad (3.34)$$

$$Re = \frac{0.04 \sim 0.12m \cdot 0.6 \sim 1.2m/s \cdot 1000kg/m^3}{10^{-3}Pa \cdot s} = 2.40 \cdot 10^4 \sim 1.44 \cdot 10^5 \quad (3.35)$$

であり、明らかに乱流であることが分かる。径深と流速は本論文におけるシミュレーションで見られる範囲を代入した。よって本論文で行った実験では水深、速度が著しく遅い場合\*3を除き、基本的に乱流の流れとして議論する。

よって本実験の設定では定常流でかつ水流の流れは基本的に乱流だと考えられるので、常流、射流、跳水という水面の状態変化が重力勾配雑音に与える影響が興味の対象となる。特に跳水は水面を大きく揺らすため注目すべき現象である。なお、[18] によれば常流、射流、跳水と等流の発生は無関係である。

\*3 乱流でなくなる境界条件 ( $Re < 2000$ ) を考えると、 $= 2.0 \cdot 10^{-3} = m \cdot V$  となる。例を考えると、流速 0.1m/s、径深 0.02m(水深約 0.03m) が当てはまる。どちらかの値がこれより小さくなると乱流から層流へ遷移していく。この例から本論文におけるシミュレーションの状況では流速も径深もかなり小さい場合でなければ層流とならないことが分かる。

### 3.3 パイプの粗さのパラメータ

他のシミュレーションに必要なパラメータについては後述するが、ここではその中で水力学や流体力学において定義されている水路（パイプなど）の粗さのパラメータを紹介する。なお、ここでは水路（開水路）において有用なものについて説明する。ここで紹介するパラメータの他に、パイプ内に水が満水の状態で水が流れる際の管の摩擦を表す「管摩擦係数」というパラメータが存在する。管摩擦係数に加えて、管摩擦係数を導出するブラジウスの式、ニクラゼの式、プラントル-カルマンの式、コールブラックの式などについては、本研究ではパイプが満水の状態を考慮する必要がないためここでは説明しない。

#### 3.3.1 マニングの平均速度公式 (マニングの粗度係数)

定常流における平均流速を求める式に

$$v = \frac{m^{2/3} i^{1/2}}{n} \quad (3.36)$$

というマニングの式（実験式）がある。ここで  $v$  は平均流速、 $n$  は粗度係数、 $m$  は径深、 $i$  は動水勾配である。粗度係数  $n$  は  $m^{-1/3} \times s$  の次元を持ち、水路の材質によって固有の値を持ち、経験的に値が判明している（付録 B）。よって粗度係数は材質ごとの粗さを示す指標として使われる。式 (3.36) は材質と動水勾配から平均速度を計算することができる有用な式であり、逆に平均流速を測ることができれば材質ごとの粗さの指標を示すことができる。付録 B において [16] で紹介されている素材ごとに値を記載した。KAGRA で使用されているパイプは、パイプの提供会社よりこの粗度係数が  $n = 0.016$  と判明している [15]。本論文におけるシミュレーションではこの値を参考に 3.3.2 で説明する相当粗度を求め、シミュレーションにおけるパイプの粗さのパラメータを設定した。

#### 3.3.2 相当粗度の計算

相当粗度とは壁面の平均的な高さを示した量である。本実験で使用したシミュレーションソフトでは壁面の粗さの情報として相当粗度をパラメータとして使用している。ここでは ([17]、P.51~P.57) を参考に相当粗度とマニングの粗度係数の関係について説明する。

まず、相当粗度の説明に必要な摩擦抵抗係数  $f$  についての表式を紹介する。管径  $D$  のパイプを流れる乱流を考える。このとき Nikuradse の実験によるとパイプの壁面摩擦抵抗係数  $f$  は

$$\text{滑面: } \frac{1}{\sqrt{f}} = 2.0 \log_{10}(Re\sqrt{f}) - 0.8 \quad (3.37)$$

$$\text{粗面: } \frac{1}{\sqrt{f}} = 2.0 \log_{10} \frac{D}{2k_s} + 1.74 \quad (3.38)$$

となる。ここで  $Re$  はレイノルズ数、 $k_s$  はパイプの相当粗度である。ここでは粗面を考えるが、粗面かつ乱流の場合  $f$  は  $D/2k_s$  によって定まる。なおここでは壁面摩擦抵抗係数については詳しく触れないが、[17] など水理学の本を参考にすると良い。

次にパイプ内の平均流速は

$$V = \sqrt{\frac{8g}{f}} \sqrt{mi} \quad (3.39)$$

で表される ([17] より)。これをマニングの式 (3.36) と合わせると、摩擦係数  $f$  は

$$f = \frac{8gn^2}{m^{1/3}} \quad (3.40)$$

と表せる ( $g$  は重力加速度)。さらにこれを整理すると

$$\frac{1}{f} = \frac{m^{1/6}}{\sqrt{8gn}} = \left(\frac{m}{k_s}\right)^{1/6} \frac{k_s^{1/6}}{\sqrt{8gn}} \quad (3.41)$$

となる。この式では摩擦抵抗係数  $f$ 、相当粗度  $k_s$ 、粗度係数  $n$  という3つの粗さに関する定数が存在している。ここで、Nikuradse の実験により式 (3.38) で  $f$  は  $D/2k_s$  によって決まっていたことに注目する。すると Nikuradse の実験結果に適合するために式 (3.41) では、どのような  $D/k_s$  に対しても  $k_s^{1/6}/(n\sqrt{8g})$  は一定である必要がある。この一定値は例をとって  $k_s^{1/6}/(n\sqrt{8g})$  について  $D/2k_s$  の関数として書き下してみると分かるが、おおよそ 7.66 になるということが分かっている。例えば、円管パイプの場合、水が満水のときを考えるので径深は  $m = D/4$  となり、式 (3.38) と式 (3.41) を使うと

$$\frac{1}{\sqrt{f}} = 2.0 \log_{10} \frac{D}{2k_s} + 1.74 = 2^{-1/6} \left(\frac{D}{2k_s}\right)^{1/6} \frac{k_s^{1/6}}{\sqrt{8gn}} \quad (3.42)$$

$$\frac{1}{7.66} \approx \frac{\sqrt{gn}}{k_s^{1/6}} = \frac{1}{2^{1/6}\sqrt{8}} \left(\frac{D}{2k_s}\right)^{1/6} \left(2.0 \log_{10} \frac{D}{2k_s} + 1.74\right)^{-1} \quad (3.43)$$

となる。これはマンニングの式 (3.36) が Nikuradse の実験式 (3.38) と適合することも表している。よって相当粗度  $n$  と粗度係数  $k_s$  の関係は

$$n = \frac{k_s^{1/6}}{7.66\sqrt{g}} \quad (3.44)$$

というマンニングストリクラー (Manning-Strickler) の式によって表される ( $g$  は重力加速度)。

KAGRA で使用されているパイプの情報 [15] を参考にすると、KAGRA 内のパイプでは  $n = 0.016$  より、相当粗度は  $k_s \approx 3.2$  mm である。本論文ではこの値を設計したパイプの相当粗度の理論値としてシミュレーションソフトに入力し、シミュレーションを行った。

### 3.3.3 壁面 (表面) 粗さ

壁面粗さとは物体の表面においてある基準面に対しての粗さの絶対値の平均値である (算術平均高さ  $R_a$ )。これは基準面をある区間の粗さの平均値とすると壁面粗さは平均偏差となるということである。本論文においては前節の通り、粗度係数からマンニングストリクラーの式を使い相当粗度を求め、それを粗さのパラメータとしている。しかし実際の壁面を計測した際には表面自体の粗さが計測されるためその値をそのまま相当粗度とすることができない。そこで壁面粗さと相当粗度の関係式が必要となる。

[19] によると通常相当粗度  $k_s$  は実験的に求められた粗度係数または摩擦抵抗係数  $f$  より求まるので直接壁面粗さから粗度係数を求める方法は確立されていない。そこで [19] では実験的に壁面粗さと粗度係数の関係について調べている。それによると壁面粗さ  $R_a$  が 0.5 mm ( $\pm 0.12$  mm) の範囲において

$$k_s = 2 \times R_a \quad (3.45)$$

となることが示されている。つまり、相当粗度は壁面粗さの約 2 倍である。そのため相当粗度の理論値 3.2 mm から計算すると、壁面粗さは 1.6 mm となる (なお、これは [19] で示される式が適用できる範囲を越えている)。

また [20] では実際の流速などを計測しマンニングの公式を用いて粗度係数を導出している。さらに、その値と実際に測った壁面の粗さから

$$k_s = 2 \times R_a \sim R_y \quad (3.46)$$

の壁面粗さと相当粗度の関係式 (仮定) とマンニングストリクラーの式を用いて導出した粗度係数の関係を調べている ( $R_y$  は最大粗さ)。それによると、前者の方法で出した粗度係数は  $k_s = 2 \times R_a \sim R_y$  の仮定で出した粗度係数の範囲

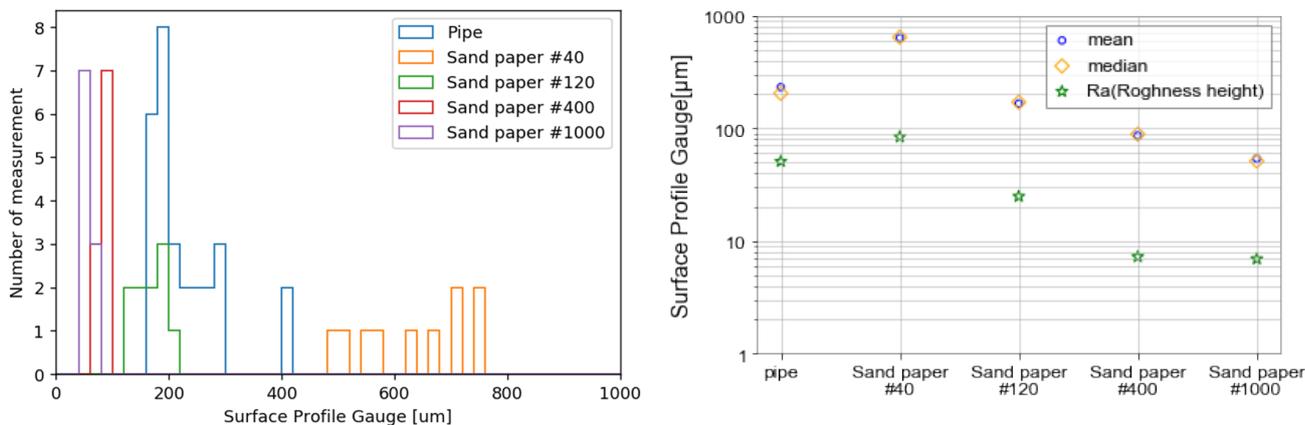
内にある。よって仮定  $k_s = 2 \times R_a \sim R_y$  がある程度正しいものであるとされる。

実際の KAGRA のパイプにおいてもこれらの関係式を参考に、実際にパイプ表面を計測して壁面粗さから相当粗度を導出した。実際の KAGRA のパイプの表面の計測は、日本ソナテスト表面輪郭ゲージ SRT-6223[21] を使用した。粗さを測る際には、表面は完全に一定の粗さを持っていないため角度や場所によって値が変化する。この測定器を使用し、測ったところ図 3.5 のようになった。図 3.5 ではパイプの表面と紙やすり (型番、40、120、400、1000) の表面を測った結果を示す。さらに、測定した表面の粗さの平均値を基準面とした時の壁面粗さを  $R_a$  でプロットした。その結果パイプの粗さの平均値や中央値は紙やすりの 120 番に近いことが分かった。紙やすりは型番が大きいくほど、壁面粗さや粗さの平均値が小さいという結果になり、型番が示す粗さの傾向と一致した。また、1000 と 400 の壁面粗さがほぼ同一になっているがこれは、[21] の分解能が  $1 \mu\text{m}$  のため、小さい粗さが測れていない可能性が考えられる。また、測定によって得られたパイプの壁面粗さは  $50 \mu\text{m}$  程度であり、最大粗さは  $R_y = 400 \mu\text{m}$  なので [19]、[20] を参考にすると、相当粗度は

$$k_s = (2 \times R_a) \sim R_y = 100 \mu\text{m} \sim 400 \mu\text{m} = 0.1 \text{ mm} \sim 0.4 \text{ mm} \tag{3.47}$$

となる。この値は粗度係数から計算される  $k_s = 3.2 \text{ mm}$  とは離れている。

本論文ではこの相当粗度 (壁面粗さ) の理論値と実際の値の誤差を考え、比較実験を行った。また、本論文では相当粗度の変化による比較以外のシミュレーションでは理論値である相当粗度  $3.2 \text{ mm}$  を採用している。実際の相当粗度を断定するのはこれからの課題としたい。壁面粗さから相当粗度を求める以外の方法としては実際のパイプ内の流速、径深、動水勾配を測定しマンシングの公式より粗度係数  $n$  を算出し、マンシングストリクターの式より相当粗度を算出する方法が考えられる。しかし、このような方法で粗度係数を測定するには手間がかかるため、壁面粗さを正確に測定し  $k_s \approx (2 \times R_a) \sim R_y$  の関係を仮定して断定するのが良いと考えられる。



(a) 各表面の粗さの頻出回数。最も粗い 40 番ではばらつきが大きい。国立天文台重力波プロジェクト 鷲見貴生氏提供。  
 (b) 各表面の粗さの平均、中央値、壁面粗さ。壁面粗さは各表面で測定した粗さの平均を基準として差の絶対値を平均した値。紙やすりの粗さの傾向は型番から考えられる粗さの傾向と一致する。

図 3.5: パイプと紙やすりの表面の粗さ。粗さの単位は  $\mu\text{m}$ 。パイプと 4 種類の紙やすりの粗さを [21] で計測した。右図 Ra は測定した粗さの平均値を基準とした壁面粗さ。

## 第 4 章

# 流体シミュレーションソフトにおける基本設定

この章ではシミュレーションにおける基本的な設定 (シミュレーションソフト上での設定) について説明する。本論文におけるシミュレーションではパラメータを変えて比較しているものがあるが、詳しい比較条件などは次章以降にて説明する。ここでは各シミュレーションに共通する設定について説明する。またシミュレーション上で再現したパイプについてもここで説明する。さらに解析する手順のうちシミュレーションソフト側の設定についてもここで紹介する。

### 4.1 流体シミュレーションソフト (Flow-3D)

本論文では流体シミュレーションソフト Flow-3D を用いて流体シミュレーションを行った。Flow-3D は flow science 社が開発した、熱流体シミュレーションソフトである [22]。本研究はこのソフトによるシミュレーションを使用した。Flow-3D は有限差分法で計算を行っており、計算の精度は有限差分法の性質からメッシュサイズに依っている。また、Flow-3D では陽解法、陰解法どちらも使用することができるが、本論文におけるシミュレーションは陽解法で行った。

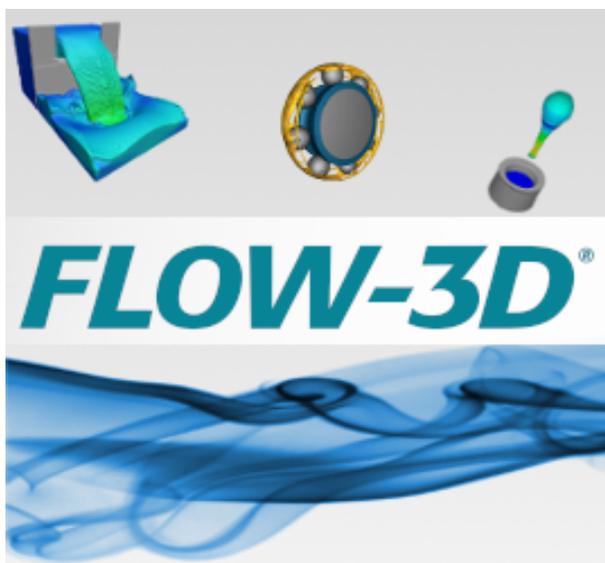


図 4.1: Flow-3D のロゴ

## 4.2 周辺の基本的環境設定

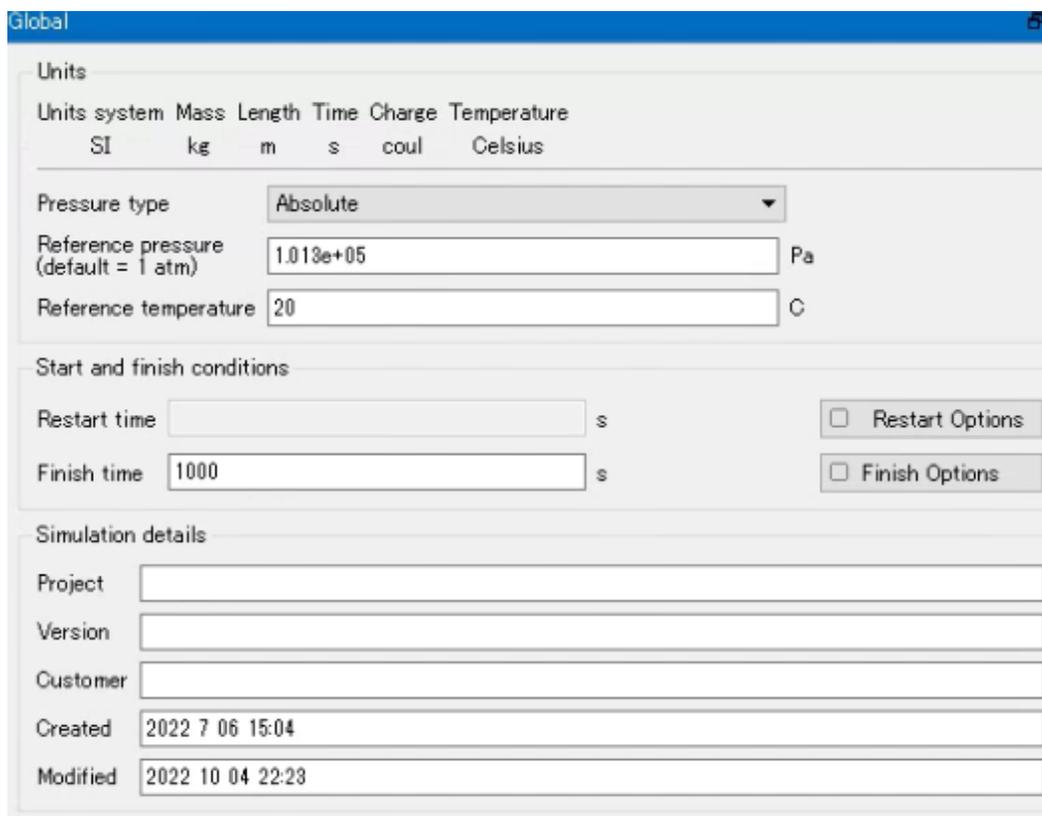


図 4.2: 環境設定、シミュレーション上の時間設定と気圧、環境温度を設定することができる。本論文のシミュレーションでは大気圧、20℃ という環境を選択した。

シミュレーション上の環境は図 4.2 のように設定した。設定した環境は、大気圧下 (1013hPa)、気温 20℃ である。KAGRA の環境を参考にすると同程度の設定で良いと考えられる。またパイプ内は密閉されておらず、入口と出口を通して外気と触れ合っており、圧力は大気圧と考えてよい。シミュレーション内時間 (シミュレーションにかかる時間ではなく、シミュレーション上で水が流れる時間) に関しては、1000 秒または、1010 秒と設定し、10 Hz 以下の重力勾配雑音の解析では 900 秒-1000 秒をサンプリング周波数 20 Hz、100 Hz までの解析では 1000 秒-1010 秒をサンプリング周波数 200 Hz でデータを取り解析している。また、重力勾配雑音の解析に使用しない 900s までの時間はデータの計量化のためサンプリング周波数を設定しない。なお、Flow-3D の仕様によりサンプリング周波数を設定しない場合でも全シミュレーション時間を 10 分割する間隔で物理量のデータが記録される。シミュレーション時間とサンプリング周波数の設定は詳しく後述する。座標は水の進行方向を  $x$  正方向、重力方向を  $z$  方向とした。

### 4.3 採用した物理モデル

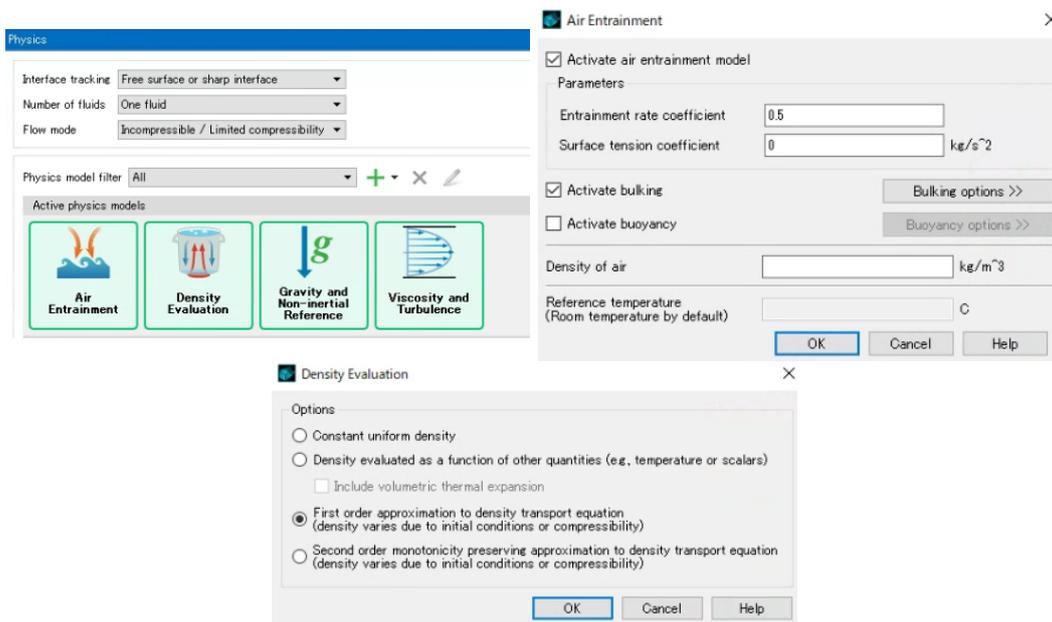


図 4.3: ここでは使用した物理モデルについて設定した。設定した物理モデルは気泡を含んだ流体を再現するモデル、重力を設定するモデル (地球の重力加速度の効果を入れるため)、流体の乱流モデルを設定した。



図 4.4: 地球の重力加速度  $9.8 \text{ m/s}^2$  を設定した。重力に対して負の方向を  $z$  軸の正方向としているため  $9.8 \text{ m/s}^2$  と設定した。

シミュレーションで使用した物理モデルは図 4.3 の通りである。Air Entrainment は気泡を伴った水の動きの表現に関するモデルである。Density Evaluation は Air Entrainment に付随するモデルでデフォルト値のまま設定してある。Gravity and Non-inertial Reference は重力の設定である。シミュレーションでは  $z$  方向に重力加速度  $9.8 \text{ m/s}^2$  を設定した。パイプの傾きの効果はオブジェクト自体を傾けることで再現している。Viscosity and Turbulence は乱流モデルの設定である。シミュレーションでは RNG model を採用した。

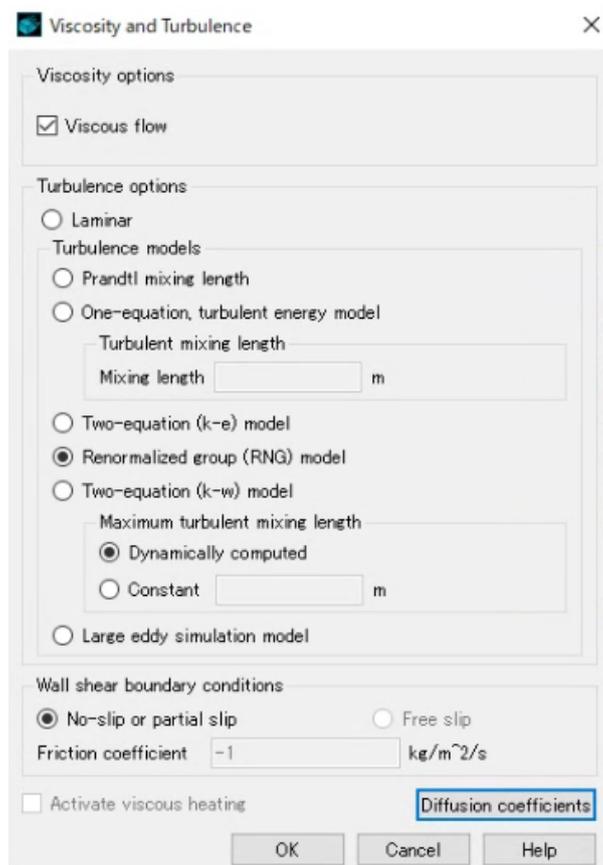


図 4.5: 乱流モデルを設定した。本論文におけるシミュレーションは水路の形状から乱流であることが予測でき乱流をよく表現できるモデルを選択した。

## 4.4 流体の設定

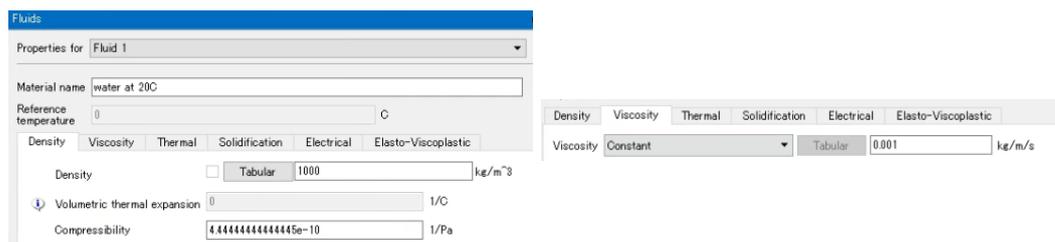


図 4.6: 流体の設定。本論文では KAGRA における地下水を対象とするため、「水」を流体として選択した。流体の密度や粘度を入力する必要があるが、ここでは水のものを入力した。

シミュレーションでは 20℃ の水を流れる流体として設定した。KAGRA のパイプで流れるのは地下水であるため、完全な純水より不純物が多いと思われるが、流体の動きを追う目的では純水が流れると考えて問題ないと考えられる。ここではさらに密度と粘性を設定した。密度は  $1000 \text{ kg/m}^3$ 、粘性は  $0.001 \text{ kg/m/s}$  である。

## 4.5 設計したパイプ

具体的なシミュレーションの目的は次章以降で述べるが、本論文で行ったシミュレーションは「直線状のパイプ」と「Y-end のパイプを再現したパイプ」を使用して行った。ここではそれぞれのパイプの設計やパイプ自体の設定について説明する。

### 4.5.1 直線状のパイプ

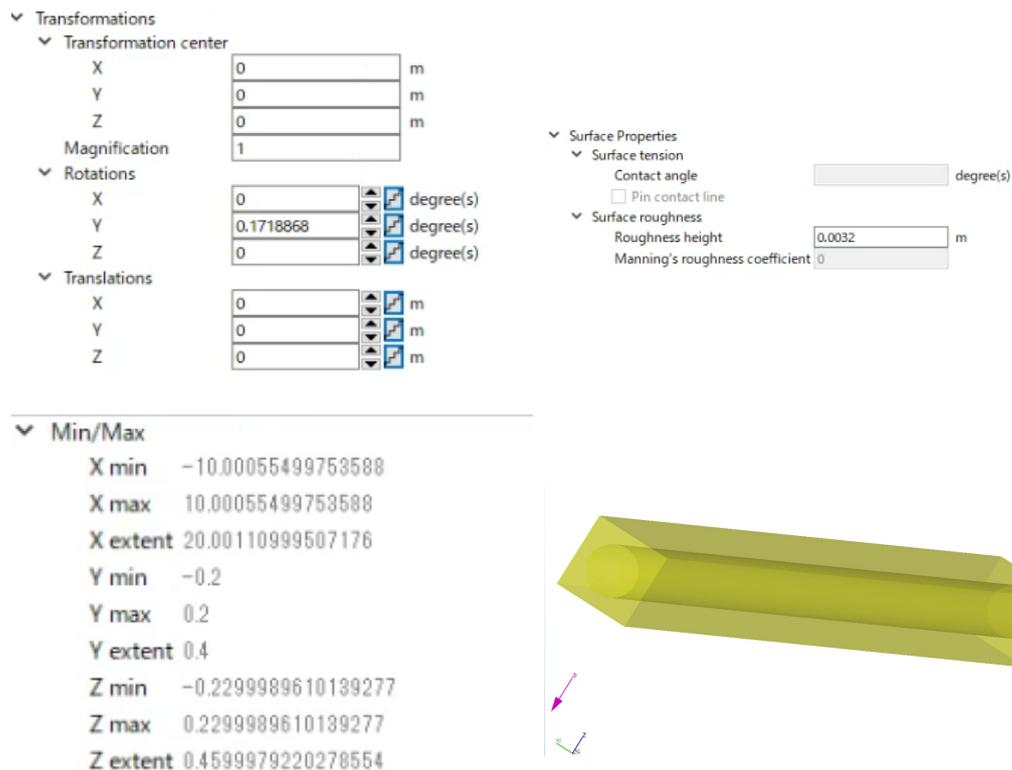


図 4.7: Flow-3D 上で設計を行った。パイプ長 20 m、半径 0.2 m の円柱パイプを設計した。傾きはここで設定している (0.3 %)。黄色く見えているのはパイプの外側の部分で円柱の空洞を囲むように設定している。これは水を入口から流す際に円柱の空洞部分にのみ水が入るようにするための措置であり、実際の KAGRA のパイプと形状が異なる。シミュレーション自体は円柱の空洞内の水の動きを再現するため、外側が直方体であることの影響はない。

直線状のパイプの設定は図 4.7 の通りである。パイプ長 20 m、半径 0.2 m の円柱パイプを設計した。なお、シミュレーション上の都合でパイプ外側は直方体のオブジェクトだが、水を流すパイプ部分は円柱上の空洞 (hole) であるため問題は発生しない。パイプの傾斜は 0.3 % であり、出口方向を下に傾いている。この傾斜は Y-end に入る前のパイプの傾斜を参考にしてしている。

## 4.5.2 Y-end のパイプを再現したパイプ

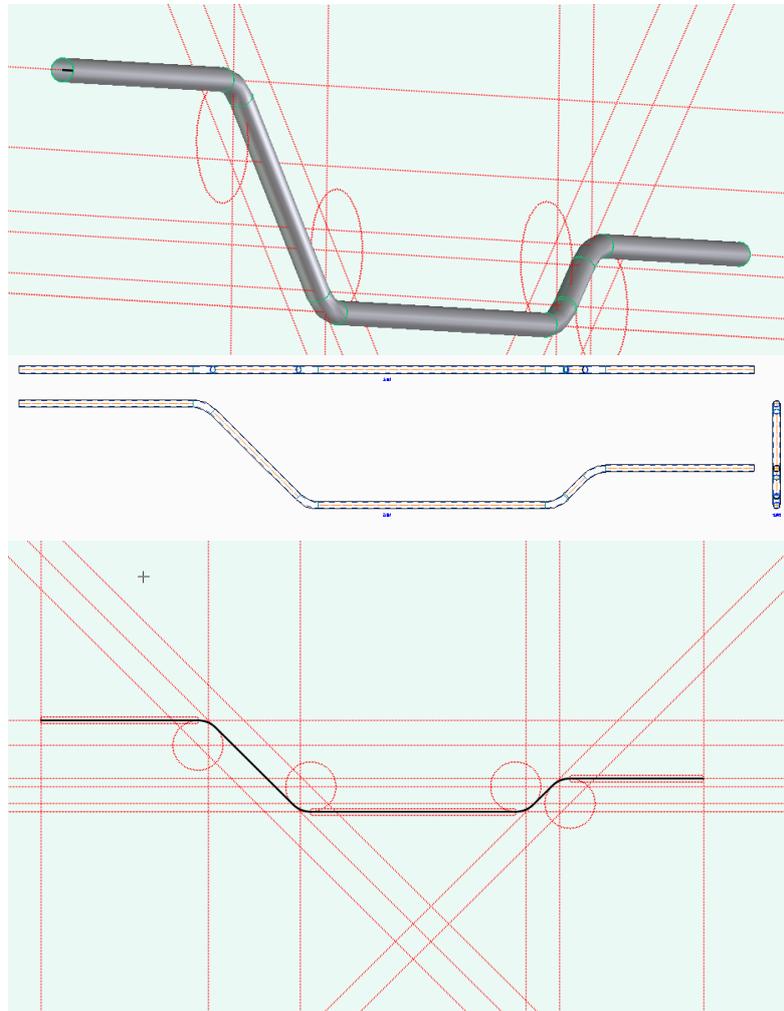


図 4.8: CAD ソフト「Creo Elements Direct/Modeling Express 8.0」上で設計したパイプ。パイプ長は  $x$  方向に 40 m、 $y$  方向に約 6 m、断面が半径 0.2 m の円となるパイプを設計した。この設計は KAGRA の Y-end のパイプの設計図を元に作成しており、実際のパイプとほぼ同じ形状のパイプを設計している。

上図:設計したパイプの概形。断面は円となっており、中は空洞となっている。中図:設計したパイプを  $x$ 、 $y$ 、 $z$  方向それぞれの方向から見た図。 $z$  方向から見るとパイプが途中で曲がっていることが分かる。

下図:パイプを作成する際に使用した補助線 (赤)。黒線がパイプの中心線であり、この中心線を元にしてパイプを作った。パイプの曲がっている部分は円を使用し、曲率を同一にしている。

Y-end のパイプを再現したパイプはまず CAD ソフト上でパイプの設計を行い、そのデータを Flow-3D で読み込みシミュレーションを行った。設計で使用した CAD ソフトは「Creo Elements Direct/Modeling Express 8.0」である [24]。この製品を使用した理由については、初学者でも扱い易い、ソフトが軽い、Y-end のパイプの設計であれば無償版で可能であるという点にある。CAD ソフトによって作成したパイプの概形と各方向から見た図面は図 4.8 の通りである。パイプ長は  $x$  方向に 40 m、 $y$  方向に約 6 m、断面が半径 0.2 m の円となるパイプを設計した。この設計は実際の KAGRA の Y-end 内のパイプの設計図を参考に作成した。しかし、パイプの曲がっている部分の曲率に関しては記載がなかったため、曲率を 0.1 としている (4 箇所曲がりの曲率は同一)。

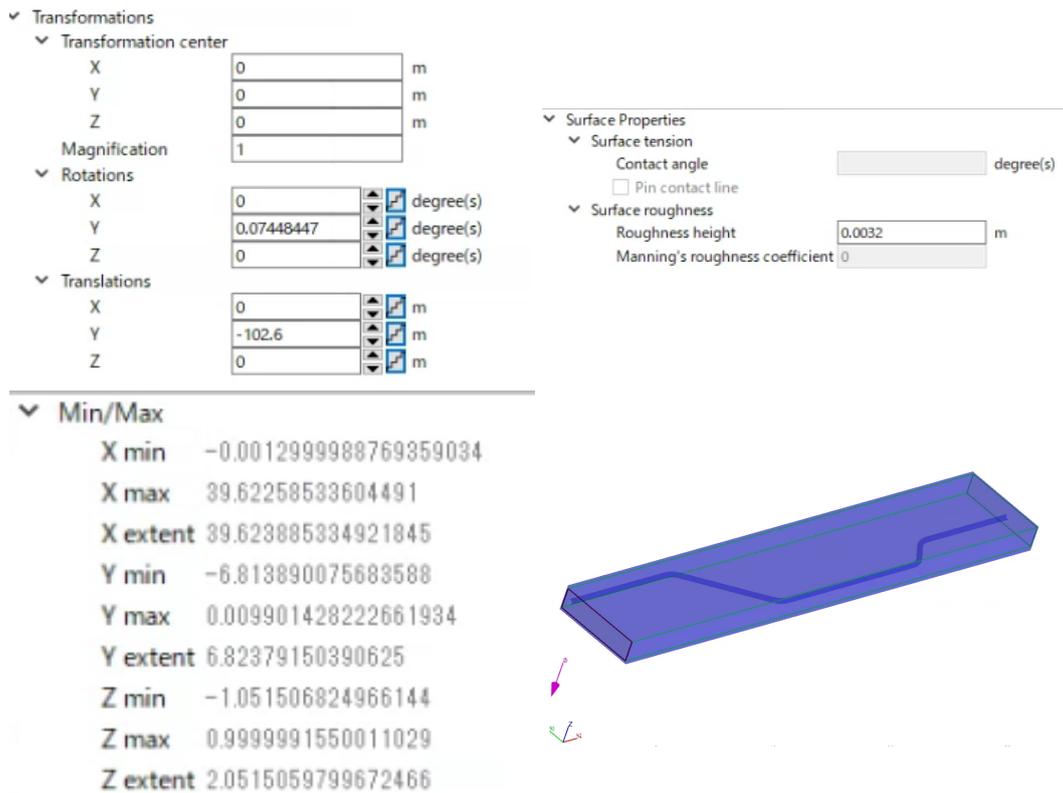


図 4.9: CAD ソフト上で設計したパイプを Flow-3D にインポートした。パイプの傾斜は Flow-3D 上で設定している (0.13 %)。水を流す都合上真っすぐなパイプと同様に直方体のオブジェクトで覆っているが、パイプ部分は空洞になっており問題なくシミュレーションが行える。

Y-end のパイプを再現したパイプの設定は図 4.9 の通りである。図 4.8 で設計したパイプを Flow-3D で読み込み、適切な座標に配置、傾斜を設定したものである。設定の都合により原点はパイプの入り口付近となっている。テストマスは正しい位置関係となる座標に設定し計算した。Flow-3D ではさらに相当粗度を設定している。なお、シミュレーション上の都合によりパイプの外の空間を直方体のオブジェクトにしているが、水を流すパイプ部分は空洞であるため問題は発生しない。パイプの傾斜は 0.13 % であり、出口方向を下に傾いている。この傾斜は Y-end に入る前のパイプの傾斜を参考にしており、この設計パイプは完全に滑らかな円ではない。CAD ソフトの性質上、完全に滑らかな曲線は作成できず、円に近い多角形となっている。

## 4.6 メッシュと境界条件

1つのメッシュの形は直方体である。またメッシュ領域も直方体状にのみ設定できる。メッシュ領域は  $x$ 、 $y$ 、 $z$  面ごとに境界条件を設定することができ、一つのシミュレーションで複数のメッシュ領域を設定することもできる。

### 4.6.1 直線状のパイプ

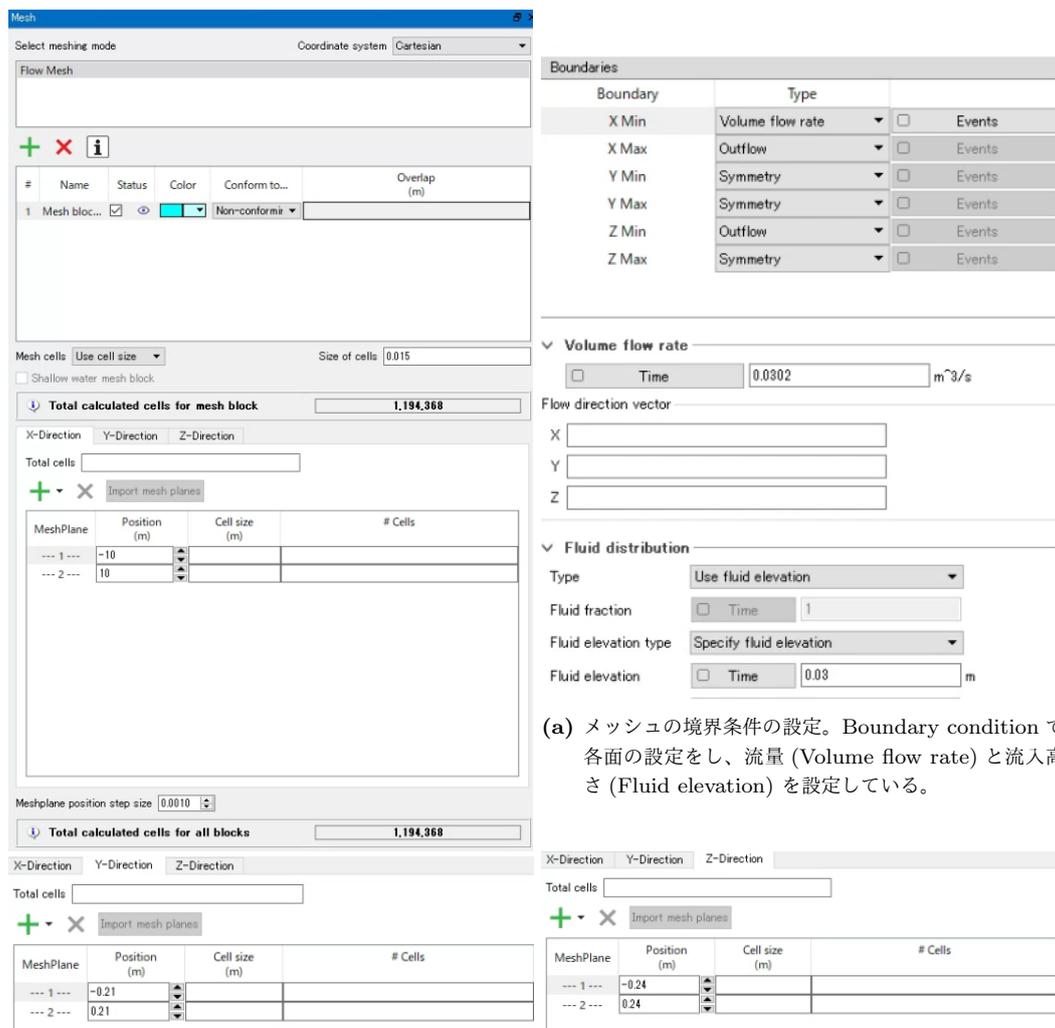


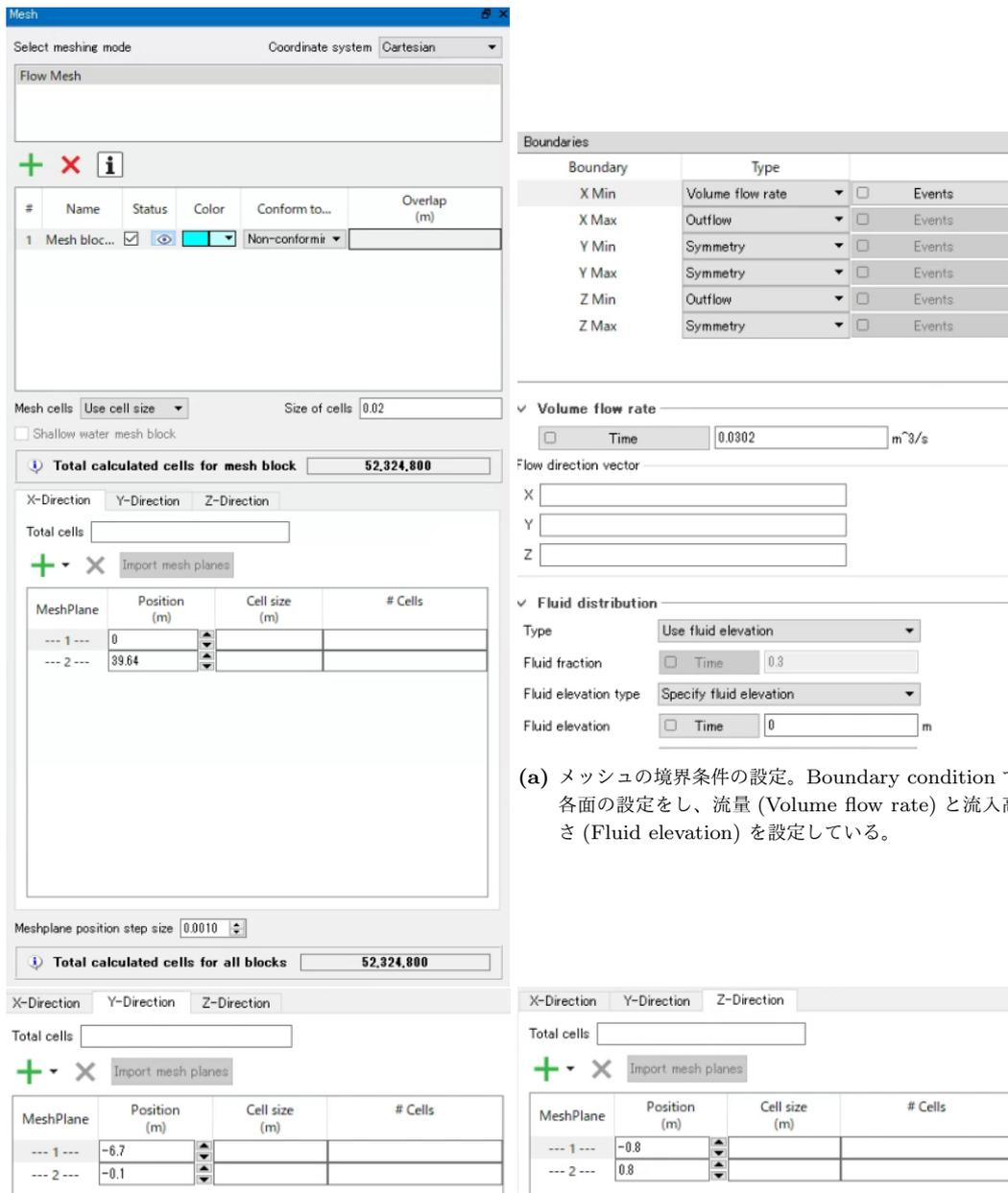
図 4.10:  $x$  方向に  $-10\text{ m} \sim 10\text{ m}$ 、 $y$  方向に  $-0.21\text{ m} \sim 0.21\text{ m}$ 、 $z$  方向に  $-0.24\text{ m} \sim 0.24\text{ m}$  と設定した。これらは 1 つのメッシュ領域で設定している。この設定はパイプの空洞部分を覆うような設定であり、パイプの全ての流体の動きを追うことができる。

メッシュ領域は円柱部分に合わせて設定した。具体的には  $x$  方向に  $-10\text{ m} \sim 10\text{ m}$ 、 $y$  方向に  $-0.21\text{ m} \sim 0.21\text{ m}$ 、 $z$  方向に  $-0.24\text{ m} \sim 0.24\text{ m}$  と設定した (図 4.10)。メッシュ領域の形状は直方体で円柱部分が内包されるように設定した。メッシュサイズはメッシュサイズの比較実験により  $1.5\text{ cm}$  と設定した。

メッシュの境界条件について説明する。境界条件は直方体メッシュの  $x$ 、 $y$ 、 $z$  方向に垂直なそれぞれの面に対して設定することができる。 $x$  最小面 (パイプ入口) では指定した量の水が指定した高さから流れ出るように設定している。 $x$

最大面 (出口) では outflow と設定し、水がそのまま排出する設定としている。入口の設置の高さは  $z$  座標の値であり、入力した座標より小さい  $z$  座標の地点全てに水が流入する。本論文のシミュレーションではどの流量であってもパイプの断面の中心と同高さから水が流れるように設定した (流入量が小さければ、水の初速度が遅くなる)。直線状のパイプではパイプ中心を支点として傾斜をつけているため、入口では  $z = 5\text{ cm}$  地点がパイプの中心の高さとなる。他の面に関しては水と接触しないため特に境界条件は設定していない。

### 4.6.2 Y-end のパイプを再現したパイプ



(a) メッシュの境界条件の設定。Boundary condition で各面の設定をし、流量 (Volume flow rate) と流入高さ (Fluid elevation) を設定している。

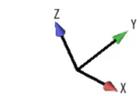
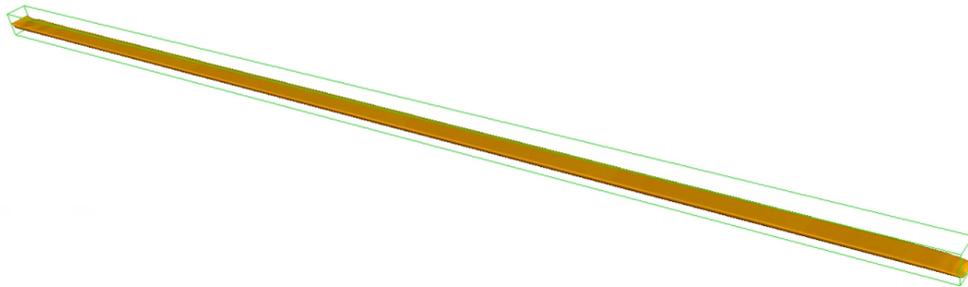
図 4.11:  $x$  方向に  $0\text{ m} \sim 40\text{ m}$ 、 $y$  方向に  $-0.1\text{ m} \sim -6.1\text{ m}$ 、 $z$  方向に  $-0.8\text{ m} \sim 0.8\text{ m}$  と設定した。これらは全て 1 つのメッシュ領域で設定している。曲がったパイプにおいては水が通らない場所までメッシュ領域を設定しているが、これは 1 つのメッシュ領域で空洞部分を覆うためである。

メッシュ領域はパイプの外の空間を埋めるオブジェクトに沿って直方体状に設定した。具体的には  $x$  方向に 0 m ~ 40 m、 $y$  方向に -0.1 m ~ -6.1 m、 $z$  方向に -0.8 m ~ 0.8 m (図 4.11)。メッシュサイズは 2 cm とした。詳細については後章に記載する。他に、データサイズを小さくするためにメッシュ領域をパイプに合わせ (直線部部分と斜め部分と分けて) メッシュ領域を設定することができる。しかし、Flow Science 社より流体が流れる途中でメッシュの境界を跨ぐと、境界条件により高精度なシミュレーションを目的とした場合現実の挙動から外れてしまう可能性もあるという指摘を受けた。そのため本シミュレーションではメッシュの境界を通過すること避けるために一括でメッシュ領域を設定している。

メッシュの境界条件は基本的には真っすぐなパイプと同様に設定した。入口の水の流入の高さのみ Y-end のパイプを再現したパイプのシミュレーションでは、原点がパイプ入口付近のため、流入高さを  $z = 0$  m としている。

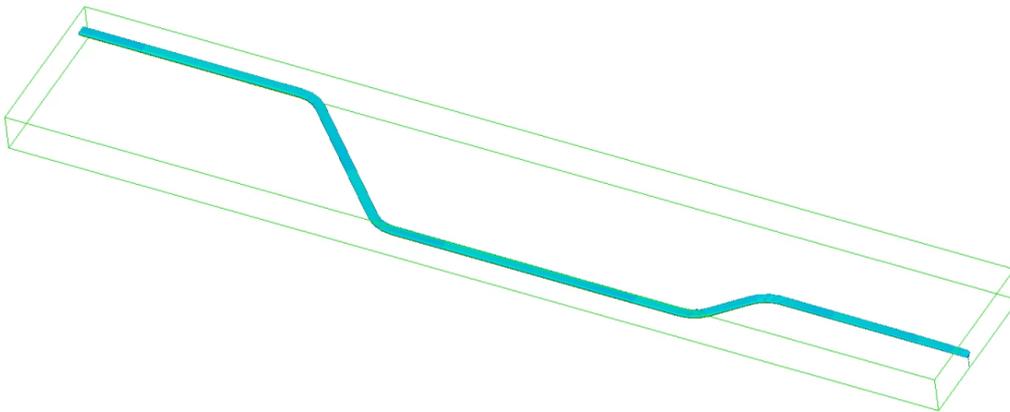
## 4.7 流体シミュレーションの視覚的な様子

Time = 1000.000183



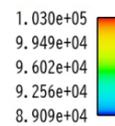
FLOW-3D

Time = 1010.00244



FLOW-3D

Pressure (Pa)



Pressure (Pa)

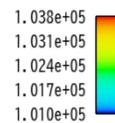


図 4.12: Flow-3D に付随するソフト「Flow sight」を使用し、Flow-3D のシミュレーションの様子を観察した。ここでは流体のみ表示されるように設定してある。水が設定したパイプの入り口断面から流入し、パイプに沿って流れ、パイプ出口から放出されていることが分かる。

シミュレーションの内容や結果については次章以降に説明するが、基本的なシミュレーションの様子をここで示す。使用する Flow-3D とは別に存在する”flow-sight”というソフトを使用すると、Flow-3D で行ったシミュレーションの様子を視覚的に見ることができる。本論文で行ったシミュレーションでは基本的に図 4.12 のように水が流れている。

## 4.8 シミュレーションソフトの仕様に起因する回避した設定

本論文でシミュレーションをするにあたり、いくつか回避した設定が存在する。本文中では詳しく触れないが、簡単に説明する。まず、メッシュ領域を複数設置する設定である。これは先述したが、メッシュ領域を流体が跨ぐ際に、境界条件によって自然な流れが再現されない場合があるからである。そのため、本論文におけるシミュレーションは全て単一のメッシュ領域を設定しそのメッシュ領域の端にのみ境界条件を設定している(メッシュ領域の端は水の流入口と出口)。

次にメッシュサイズを途中で変化させる設定である。Flow-3D ではシミュレーションを中断し、その地点から設定を変更しシミュレーションを再開させることができる。この機能を利用し、900 秒までをメッシュサイズ 2 cm、900 秒から 1000 秒をメッシュサイズ 1 cm でシミュレーションを行った。このように途中までを粗いメッシュにすることでシミュレーションのサイズや時間を短縮できると考えたからだ。しかし、このシミュレーションの 900 秒から 1000 秒の部分を 1000 秒をメッシュサイズ 2 cm で行ったシミュレーションの 900 秒から 1000 秒の部分と比較したところ、結果(重力勾配雑音)に大きな差異があった。これは途中まで粗いメッシュにするとそこまでのシミュレーションの誤差が大きくなるといったことが考えられる。さらに水面の様子を調べてみると、前者のシミュレーションでメッシュサイズを変更したことが原因と考えられる現実には発生しない(現実ではメッシュサイズの変更ではなく、静的な状態であるため)波が発生していることが確認できた。そのため、途中でメッシュサイズを変更してしまうと、誤差が多すぎることや現実にはない波が発生してしまうため、途中でメッシュサイズの変更は行わずシミュレーションをおこなった。

つぎに回避した設定はシミュレーションを中断して再開することである。Flow-3D にはシミュレーションを一旦終わらせた後に、そのシミュレーションを設定を変えて再開できる機能がある。この機能を使用し、900 秒でシミュレーションを一旦終わらせ、900 秒から 1000 秒を新しいシミュレーションとして”設定を変更せずに”再開したシミュレーションと 1000 秒を通してシミュレーションしたものを比較した。この比較では「メッシュサイズが大きい」かつ「流量が多い」条件で前者のシミュレーションを行うと、再開した 900 秒以降のシミュレーションで不自然に急激に水面が揺れなくなる現象が発生した。同時間の 1000 秒通したシミュレーションを見てみると、そのような急激に水面が揺れなくなる現象は発生していないため「メッシュサイズが大きい」かつ「流量が多い」条件でシミュレーションを分割して行う設定は避けた。なお直線状のパイプのシミュレーションをメッシュサイズ 1.5 cmで行っているが、1.5 cm の場合どの流量でも分割してシミュレーションしても一括でシミュレーションした場合と比べて変化があまりなかった。そのため、本研究の時間的制約の関係上、初期流量の比較のシミュレーションのみ 1000 秒と 10 秒でシミュレーションを分割して結果を出している(この機能の検証により、通して 1010 秒のシミュレーションを行った結果と変化がないことが分かっている。)

最後に、避けた設定ではなく、避けずに問題が発生しなかった設定を紹介する。前述の通りメッシュサイズの変更はシミュレーションに影響を与えたが、サンプリング周波数の変更は影響を与えなかった。Flow-3D の設定により、シミュレーションを途中で終わらせずに、サンプリング周波数のみ途中で変更することができる。(シミュレーション時間上の途中という意味でありあらかじめシミュレーションの前に時間ごとのサンプリング周波数を設定する必要がある。シミュレーションが始まった後に途中から変更することはできない)さらに、サンプリング周波数の変更がシミュレーション自体に影響を与えないことは Flow science 社より確認が取れている。そのため、本論文におけるシミュレーションは途中でサンプリング周波数を変更する設定を採用している。

## 4.9 Flow-3D からの出力

ここではシミュレーション自体の設定ではなく、Flow-3D からのデータの出力について説明する。この節で説明するデータやその使用範囲は Flow-3D でデータを出力する際に Flow-3D のインターフェイス上で設定している (つまり Flow-3D の機能によりデータの範囲を設定している)。Flow-3D からの出力では出力するパラメータ、出力する時間範囲、出力する範囲 ( $x, y, z$  の三次元) を設定し、テキストデータで出力することができる。この Flow-3D からの出力されたデータを Python を用いて解析するが、その解説は 7 章で行う。

### 4.9.1 使用したデータ

重力勾配雑音の計算において使用したデータは水深 (flow depth)、パイプの地形データ (terrain elevation) の 2 つである。それぞれ flow3D のデータ出力機能を用いてテキストデータで出力することができる。flow3D のマニュアルを参考に各パラメータについて説明する。

flow depth はメッシュにより設定された各  $x, y$  座標における水深の値である。水深はその座標にある流体の体積をメッシュによった設定された範囲の底面積で割った値である。flow3D の出力の仕様で同じ  $x, y$  座標であれば同じ値を持つ ( $z$  方向に対して同じ値が記録される)。

terrain elevation は設定したメッシュ内の  $z$ -方向から見た時のオブジェクトとオブジェクトがない領域の境界の座標である。本論文におけるシミュレーションのメッシュの設定ではパイプの下側壁面の座標である。

#### テキストデータの構造



図 4.13: テキストデータの構造

flow depth のテキストデータの構造は図 4.13 の通りである。flow depth は時系列データであるため、各時刻におけるデータが連続で記録される。さらに、時刻のデータの中にその時刻における各地点の水深のデータが記録される。このような構造のテキストデータを配列として読み込みプログラムで処理することとした。

#### 4.9.2 重力勾配雑音の計算に使用する部分の設定

前節で説明したように Flow-3D から出力することができる水深 (flow depth) とパイプの地形データ (terrain elevation) のデータを使用し重力勾配雑音を計算する。ここではこれらのデータの使用範囲 (積分範囲) について説明する。式 (1.32) より重力勾配雑音の計算には水の質量の足し合わせを行う。万有引力は距離の逆 2 乗に比例し小さくなるが完全には無くなることはないので、パイプ内の水からの万有引力の足し合わせを計算する場合には、理想的には十分に長いパイプの領域と時間 (さらにメッシュサイズの細かさ) が必要である。しかし、シミュレーションはデータ量やシミュレーション時間により制限を受け、重力勾配雑音の計算においても限定的な範囲の積分を行うことになる。本論文で行う解析では可能な限り、実際の重力勾配雑音に近いものを推定するために、これらの制限を受け入れながら影響を最小限にすることを考えた。

##### 使用する時間

重力勾配雑音の計算で使用するシミュレーション時間について説明する。また、選んだデータの処理については後述する。本論文におけるシミュレーションは水の流れる時間を 1000 秒または 1010 秒の設定にして行っている。本論文では水が収束するまでの十分な時間として 900 秒を取った。本論文のシミュレーションは水がない状態のパイプに水を流し、シミュレーションを行う。流入量が一定の場合、水が収束するまでには、① 水がパイプ出口に到達する、② 流入と排水のバランスがとれ水面が安定する、という大きく 2 つの状況を経る。本論文のシミュレーションではこの 2 つの状況を前提に、まず 900 秒を水面が安定するまでに十分に長い時間だと仮定してシミュレーションを行い、900 秒までに②の状況になっているかを確認した。ここでは例に直線状のパイプにおけるシミュレーション (初期流量 109 t/h、相当粗度 3.2 mm) における水面の収束の確認を示す (図 4.14)。図 4.14 では 100 秒時にすでに水面 (水深) の分布が 1000 秒時の分布とほぼ同じ形になっている。よってこの条件では 100 秒ですでに水面が収束していると言える。このように、本論文におけるシミュレーションでは 100 秒前の早い段階で全体的には水面が収束しているが、乱流のため局所的に完全に水面が落ち着くことはない。また、この例から 1000 秒というシミュレーション時間は十分に長いシミュレーション時間と言える。

次にサンプリング周波数だが先述の通り、900 秒までをサンプリング周波数設定なし、900 秒～1000 秒をサンプリング周波数 20 Hz、1000 秒～1010 秒をサンプリング周波数 200 Hz で行っている。つまり重力勾配雑音の計算では 10 Hz 以下の重力勾配雑音を評価する場合は 900 秒～1000 秒の水面データ、100 Hz 以下の場合は 1000 秒～1010 秒の水面データを用いる。このように 100 秒間と 10 秒間の区間に分けている理由としてデータ量の制限がある。サンプリング周波数が高いほどその分データ量は増えるので、例えばサンプリング周波数 200 Hz で 100 秒間のデータを取ろうとするとサンプリング周波数 20 Hz のデータ量と比べて 100 倍増えることとなり膨大なデータ量となる。なお、計算に使用する時間に関しては真っすぐなパイプと Y-end のパイプを再現したパイプのシミュレーションどちらの場合においても、同様の設定でサンプリング周波数の設定と解析を行った。

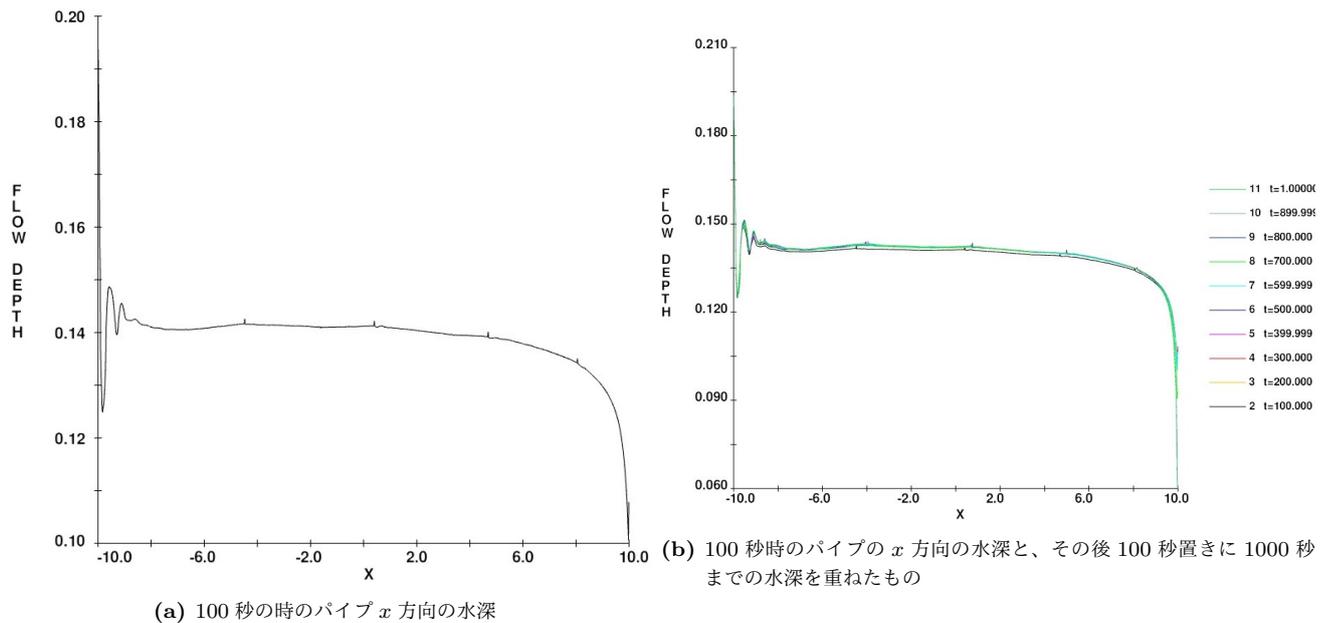


図 4.14: 直線パイプにおけるシミュレーション (初期流量 109 t/h、相当粗度 3.2 mm) においてパイプ中心線 ( $y = 0$  m) に沿った  $x$  方向の水深。縦軸は水深で横軸は  $x$  方向である。100 秒時ですでに 1000 秒時の水深の分布とほぼ同じ分布になっていることが分かる。

#### 使用する水面の範囲

重力勾配雑音の計算で使用する水面の領域の範囲について説明する。水面の範囲については直線状のパイプと Y-end のパイプを再現したパイプで条件が違うのでそれぞれ説明する。なお、計算に使用する水深は 2 次元データであるため勾配については言及しない。

直線状のパイプはパイプ長 20 m である。重力勾配雑音の計算ではその中の入口と出口からそれぞれ 4 m を除いた 12 m の水面を使用している。また  $y$  方向 (横方向) には水面が存在する部分の全領域を積分範囲としている。これはパイプの入口と出口で起こる現象は条件によって異なる場合が多く、さらに直線状のパイプのシミュレーションではパイプ内の一様な水の流れを作る重力勾配雑音に注目しているため入口や出口の効果を除くためである。また、12 m の領域の設定は本論文における直線状のパイプのシミュレーションではパイプ中心を原点とした場合、鏡は (0 m、5 m、1.5 m) の位置にあるためである。つまり、鏡とパイプの近い場所が 5 m~6 m 離れているため、同スケールの 6 m を  $x$  座標 0 地点から取る措置である。

次に Y-end のパイプを再現したパイプは  $x$  方向には約 40 m、 $y$  方向には約 6 m である (図 4.8、図 4.9 を参考)。計算に使用した部分は入口と出口からそれぞれ 5 m の部分を除いた  $x$  方向に 30 m の部分である。また、 $y$  方向にはパイプ内の水面がある領域全てを積分範囲とした。入口と出口を除いた理由は真っすぐなパイプのシミュレーションと同様である。

## 第5章

# 直線状パイプにおける地下水のシミュレーション

この章では直線パイプのシミュレーションの目的と設定内容について説明し、シミュレーションを解析して得られた水面のゆらぎや水流についての考察を行う。解析の結果得られた重力勾配雑音については7章で考察する。本論文ではKAGRAのパイプ配置における地下水シミュレーションを行う前に真っすぐなパイプのシミュレーションを行った。これは、曲がり角のない単純な直線パイプでパラメータ変化させることにより、曲がり角の影響などを考えずにパラメータが変化したこと由来するパイプ内の水のゆらぎの変化や重力勾配雑音の変化の範囲についての比較を行えるためである。その後、直線パイプの考察で得られた知見を元にしてKAGRAのパイプ配置における地下水のシミュレーションの水面の様子や重力勾配雑音を考察した。

### 5.1 シミュレーションの内容

パイプを流れる水の基本的な挙動を見るために、パイプ長20m、内径0.4mの真っすぐなパイプでシミュレーションを行った。また、水の動きを見るとともに、パイプを流れる水が鏡(質点)に与える重力勾配雑音の影響についても解析を行った。本論文ではメッシュサイズ、初期流量、相当粗度、傾斜、それぞれ条件を変え比較を行った。ここではそれぞれの比較の目的について説明する。

まずメッシュサイズの比較のシミュレーションだが、この比較は正しいシミュレーションをすることができるシミュレーションの精度を調べるために行った。正しいシミュレーションとは同じ状況の実際の物理現象を正しく再現できる状態のことである。正しいシミュレーションの基準は流体シミュレーションでは、シミュレーションの精度を上げていき(メッシュサイズを小さくしていく)、見たい物理量の値がある精度からそれ以上精度を上げて変化しなくなった時とする。本論文で見たい物理量は、重力勾配雑音であるので、メッシュサイズを小さくしていき重力勾配雑音に変化しなくなる条件を調べた。なお、シミュレーションの精度(メッシュサイズの小ささ)とシミュレーション時間はトレードオフの関係にあり、現実的なシミュレーション時間に収まる範囲まで精度を上げる。本論文ではこのメッシュサイズの比較より決定された最も良い精度のメッシュサイズを用いて、直線パイプのシミュレーションを行った。なお、KAGRAのパイプ配置における地下水のシミュレーションではシミュレーションの大きさなどが異なるため、メッシュサイズの比較で決定したメッシュサイズを前提としつつ、シミュレーション時間を考慮して再度メッシュサイズの設定を行った。

次に、初期流量を変化させたシミュレーションの比較の目的について説明する。この比較はKAGRAにおいて季節の変化などにより地下水の量が増減し、パイプ内の流量が増減するため、その変化がどのように水面や重力勾配雑音に影響するかを調べるために行った。比較に用いた初期流量の設定は現実のKAGRAのパイプにおいて現実的な水量

を設定した(図5.1)。図5.1は対象とする中央排水管だけでなく強制排水管も含めた流量の測定値である。これを見ると、流量が多い時期で $350\text{ m}^3/\text{h}$ 、流量が一時的に0になる9月のデータを除けば、少ない時期で $100\text{ m}^3/\text{h}$ である。強制排水管との合計流量であることを考えると、中央排水管の流量はこの水量の2~3分の1程度であることが見積もられ、その流量の範囲は $33\text{ m}^3/\text{h} \sim 175\text{ m}^3/\text{h}$ 程度である。

図5.1: 実際のKAGRAのパイプの流量(中央排水管と強制排水管の合計の流量)。2021年5月から2022年月までの測定値。国立天文台重力波プロジェクト鷲見貴生氏提供。

なお、流量の変化は実際のKAGRAのパイプにおいて唯一の変化パラメータであり、KAGRAのパイプ配置における地下水のシミュレーションにおいても初期流量を変えて比較した。

次に、相当粗度を変化させたシミュレーションの比較の目的について説明する。パイプの相当粗度は3.3.3節において、パイプの粗度係数から計算される相当粗度が $3.2\text{ mm}$ であるのに対して粗さの実測値から算出された相当粗度が $0.1\text{ mm} \sim 0.4\text{ mm}$ であり、理論値と実測値で差があることを紹介した。ここで紹介したように実際のKAGRAのパイプの相当粗度を断定することはできなかつたので、直線パイプにおいて現実のKAGRAのパイプであり得る範囲で相当粗度の値を変化させ、重力勾配雑音の大きさの範囲を調べることを目的とした(相当粗度を断定できない以上、重力勾配雑音が極端に大きくなる条件などの特殊な条件がないかを調べる必要がある)。

次に傾斜の違いによるシミュレーションの比較の目的を説明する。2章で説明したように、KAGRAのパイプには2つの傾斜のパイプが存在する。Y-endに入る前までのアームに沿った部分の直線状のパイプは傾斜 $0.3\%$ であり、Y-end内のパイプの傾斜は $0.13\%$ である。Y-endのパイプを再現するのであれば傾斜は $0.13\%$ であるが、Y-endに入る前までの傾斜 $0.3\%$ のパイプの距離が長く、そこでゆらぎが大きい場合その影響が重力勾配雑音に現れることが考えられる。よって本論文における直線状のパイプのシミュレーションでは傾斜 $0.3\%$ でシミュレーションを行った。一方、KAGRAのパイプ配置におけるシミュレーションでは傾斜 $0.13\%$ でパイプを再現する必要がある。そこで直線状のパイプにおいて傾斜 $0.13\%$ の条件と傾斜 $0.3\%$ の条件で比較することで、水面の状況がどのように変化するかを調べ、傾斜 $0.13\%$ 特有の現象などが起こらないかを確かめた。この確認を行うことでKAGRAのパイプ配置におけるシミュレーションで傾斜が $0.13\%$ になったことによる影響がないかを考察することができる。

表5.1では直線状パイプで行ったシミュレーションの条件をまとめて示した。便宜的にシミュレーションに番号を付けている。

表 5.1: 直線状パイプのシミュレーションにおける設定パラメータまとめ

番号	メッシュサイズ [cm]	傾き [%]	シミュレーション時間 [s]	相当粗度 [mm]	初期流量 [t/h]
S1	1.5	0.3	1010	3.2	109
S2	2	0.3	1000	3.2	109
S3	2.5	0.3	1000	3.2	109
S4	3	0.3	1000	3.2	109
S5	3.5	0.3	1000	3.2	109
S6	4	0.3	1000	3.2	109
S7	1.5	0.3	1010	3.2	181
S8	2	0.3	1000	3.2	181
S9	2.5	0.3	1000	3.2	181
S10	3	0.3	1000	3.2	181
S11	3.5	0.3	1000	3.2	181
S12	4	0.3	1000	3.2	181
S13	1.5	0.3	1010	3.2	36.2
S14	1.5	0.3	1010	3.2	72.4
S15	1.5	0.3	1010	3.2	145
S16	1.5	0.3	1010	0.1	109
S17	1.5	0.3	1010	0.4	109
S18	1.5	0.3	1010	1.2	109
S19	1.5	0.3	1010	2.0	109
S20	1.5	0.13	1010	3.2	109

### 5.1.1 メッシュサイズの比較

基本的に流体シミュレーションでは設定したメッシュサイズが小さく、細くなるほどより現実に近い物理現象を表現できる。Flow-3D は有限差分法のシミュレーションであることから、有限差分法の原理的な要因でメッシュサイズが大きいほど誤差が大きくなる。しかし、適切なメッシュサイズはシミュレーションにおいて観察したい物理量や環境によって変化する。まず、本研究の目的である重力勾配雑音の推定のために適切なメッシュサイズを、メッシュサイズを変化させいくつかシミュレーションを行い、探索した。なお、メッシュサイズとシミュレーションの実行時間はトレードオフの関係にあり、現実的な時間内にシミュレーションを終わらせることと、メッシュサイズの細かさによる精度向上を目的として、適切なメッシュサイズを総合的に判断した。

#### 初期流量 109 t/h

まず、初期流量 109 t/h の条件でメッシュサイズの違いによる比較を行った。表 5.1 では番号 S1～S6 のシミュレーションの比較である。設定したメッシュサイズは、4 cm、3.5 cm、3 cm、2.5 cm、2 cm、1.5 cm である。最も細かいメッシュサイズを 1.5 cm としたのは、1 cm の条件では現実的に終わらされるシミュレーション時間とならないと予測されたからである。しかし、1.5 cm であっても大幅にシミュレーション時間は増えている。

#### 初期流量 181 t/h

次に、初期流量 181 t/h として、109 t/h の時と同様のシミュレーションを行った。表 5.1 では番号 S7～S12 のシミュレーションの比較である。

### 5.1.2 初期流量の違いによるシミュレーションの比較

初期流量を変化させ、水流の変化や、重力勾配雑音に与える影響の変化について調べた。表 5.1 では番号 S1、S7、S13～S15 のシミュレーションの比較である。このシミュレーションにおける初期流量とは設計したパイプ入口の境界条件において、設定する水の流入量である。この比較で行った初期流量は 36.2t/h、72.4t/h、109t/h、181t/h である。この流量は初流速 0.8m/s としたときに、パイプの断面に対して水量の割合が 10%、20%、30%、40%、50% となる時の水量である。また、この値は KAGRA において測定された地下水の流出量 (図 5.1) の値から考えても矛盾しない値である。

### 5.1.3 相当粗度 (壁面粗さ) の違いによるシミュレーションの比較

相当粗度 (壁面粗さ) を変化させ、水流の変化や、重力勾配雑音に与える影響の変化について調べた。相当粗度とは 3 章にて説明したパラメータである。表 5.1 では番号 S1、S16～S19 のシミュレーションの比較である。この比較で設定した相当粗度は 3.2mm、2.0mm、1.2mm、0.4mm、0.1mm である。3 章でも述べた通り、パイプの粗度係数から計算される相当粗度は 3.2mm であるが、壁面粗さの実測値より計算される相当粗度の値よりも大きい値である。よって 3.2mm を最大の相当粗度とし、それより低い値を比較対象とした。

### 5.1.4 傾斜の違いによるシミュレーションの比較

傾斜を変化させ比較を行った。表 5.1 では番号 S1、S20 のシミュレーションの比較である。この比較で行った傾斜は 0.3% と 0.13% である。3 章で説明したように Y-end に入る前までのパイプの傾斜は 0.3%、Y-end に入ってからパイプの傾斜は 0.13% である。真っすぐなパイプにおける比較では Y-end に入る前までのパイプを再現することを考え傾斜を 0.3% としている。ここでは真っすぐなパイプの傾斜が Y-end に入ってから傾斜である 0.13% であったときのシミュレーションを行い比較した。

## 5.2 メッシュサイズの比較のシミュレーションの解析結果

計算された重力勾配雑音を確認し、十分なメッシュサイズかどうかを確認する。シミュレーションにかかる時間を考え、完全に十分なメッシュサイズを定義すると、それ以上メッシュサイズを小さくしても見たい結果が変化しなくなるメッシュサイズの大きさである。シミュレーションにおけるメッシュサイズ決定の経緯を説明するために、ここで解析の結果得られた重力勾配雑音を示すが、計算方法などは 7 章で説明する。

図 5.2 を見ると、メッシュサイズ 4cm から 1.5cm までの条件では重力勾配雑音がどこかのメッシュサイズからか変化しないということは見られない。よってこの条件下では重力勾配雑音の計算において完全には十分なメッシュサイズを確保できていない可能性がある。またどちらの流量においてもメッシュサイズ 1.5cm の条件で重力勾配雑音は最も大きい。そのため、メッシュサイズをさらに細かくしてシミュレーションできた場合この結果より重力勾配雑音が大きくなる可能性がある。

次にシミュレーションにかかった時間を見る。図 5.3 を見ると、メッシュサイズを小さくすればシミュレーション時間が大きくなることが分かる。特にメッシュサイズを 2cm から 1.5cm にしたときにどちらの流量のシミュレーションでもシミュレーションにかかった時間が大幅に長くなっていることが分かる。さらに 1cm のメッシュサイズでシミュレーションを行ったところシミュレーション終了予測時間が大きいものとなったため現実的な時間で行えるシミュレーションの最小メッシュサイズは 1.5cm と判断した。

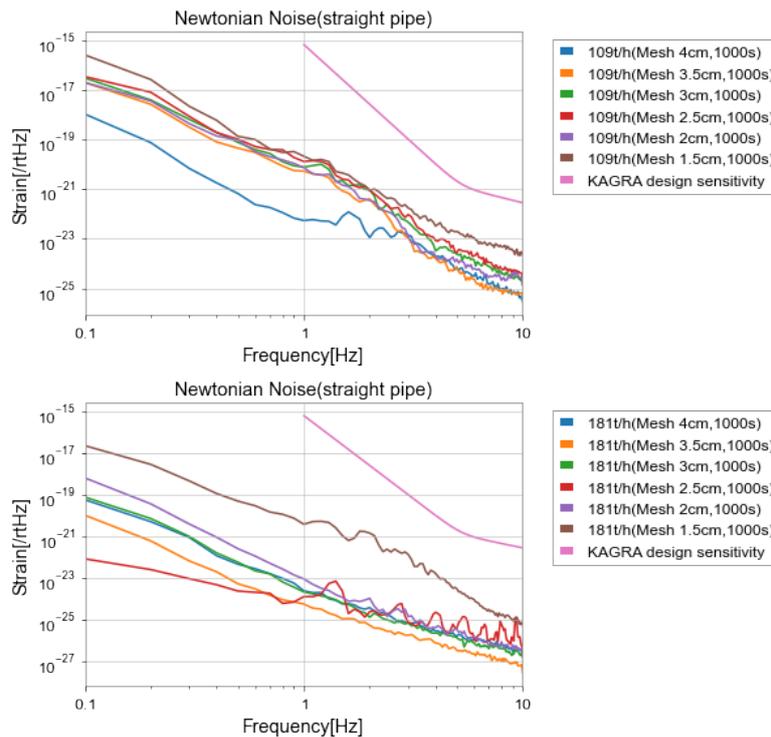


図 5.2: 上:初期流量 109 t/h でメッシュサイズを 4 cm、3.5 cm、3 cm、2.5 cm、2 cm、1.5 cm と変えて行ったシミュレーションを解析し、計算した重力勾配雑音の比較。解析には 900 秒~1000 秒の水面のデータを使用。10 秒の時間幅、5 秒間隔で高速フーリエ変換を行い平均化した。  
 下:初期流量が 181 t/h の時の比較。(シミュレーション条件、解析方法は初期流量 109 t/h の場合と同様)

よってこれらの結果から、真つすぐなパイプパイプのシミュレーションではメッシュサイズ 1.5 cm が重力勾配雑音のを計算する場合に最も良い精度ではない(図 5.2 より)が、シミュレーションにかかる時間の制約により、許容されるシミュレーション時間の中で最も細かいメッシュサイズである 1.5 cm でシミュレーションを行うことを決定した。以下の直線状パイプのシミュレーションではメッシュサイズ 1.5 cm でシミュレーションを行った。

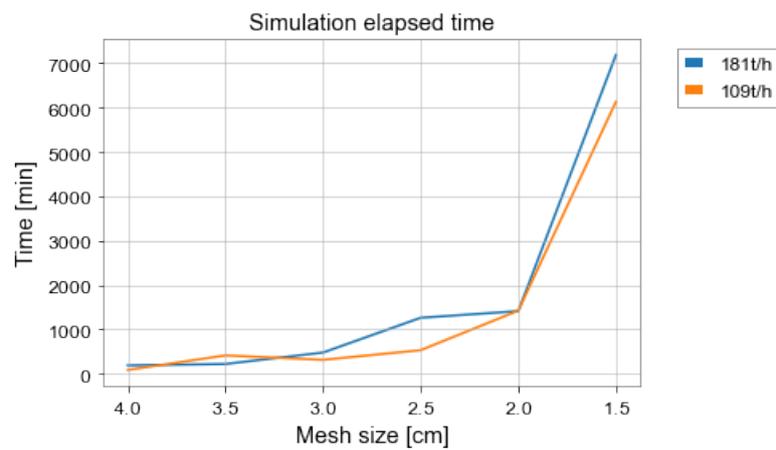


図 5.3: 各メッシュサイズごとのシミュレーション時間。同じ初期流量のデータを結んでいる。横軸はメッシュサイズ、縦軸はシミュレーション時間(分)。実際にメッシュサイズが小さくなることでシミュレーション時間が増えることが分かる。本シミュレーションの条件ではメッシュサイズ 1.5 cm となる時に大幅に時間が増える。

### 5.3 初期流量の違いによるシミュレーションの比較の解析結果

重力勾配雑音は式 (1.32) から分かる通り、質量の移動すなわち水面のゆらぎが直接影響している。よって重力勾配雑音の結果を議論するために、水面のゆらぎを調べることにした。さらにシミュレーションにかかった時間とデータサイズをまとめたものを表 5.2 にまとめた。これを見ると 109 t/h 以上の条件では 36.2 t/h と 72.4 t/h の条件と比べるとシミュレーションに倍近い時間がかかっていることが分かる。また表には秒数で記載してあるがシミュレーションに数日かかっている。データ量はシミュレーションのファイルの中の (flsgrf+ シミュレーションの名) のファイルのサイズを示している。データサイズに関しては真っすぐなパイプ全てのシミュレーションでほぼ同じであった。

表 5.2: 直線状パイプで初期流量を比較したシミュレーションのシミュレーションにかかった時間やシミュレーションのデータサイズ

番号	初期流量	シミュレーション時間	シミュレーションにかかった時間 [s]	データサイズ [GB]
S13	36.2	1000+10 秒	251680	184
S14	72.4	1000+10 秒	263240	184
S1	109	1000+10 秒	548207	184
S15	145	1000+10 秒	580686	184
S7	181	1000+10 秒	526352	184

#### 5.3.1 水面のゆらぎ

Python でプログラム (付録 C 参考) を書き、Flow-3D から出力された「Flow depth」の値を元に水面 (水深) のゆらぎを可視化した。ここではいくつかの水面のゆらぎの比較も行った。

まず、個別の地点図 5.4 を見ると、水面が落ち着いていないことが分かる。これは十分長い時間を置いても水面が収束していないということであり、水路内の流れが乱流であることを示している。よって KAGRA のパイプでは真っすぐな状態であっても、長い時間を置いても水面のゆらぎは収まらず、水の重力勾配雑音の発生は時間経過で消えないと言える。次に図 5.5 を見ると、どの条件においても、上流側の水位が高く、下流に向かっていくと水位が下がってい

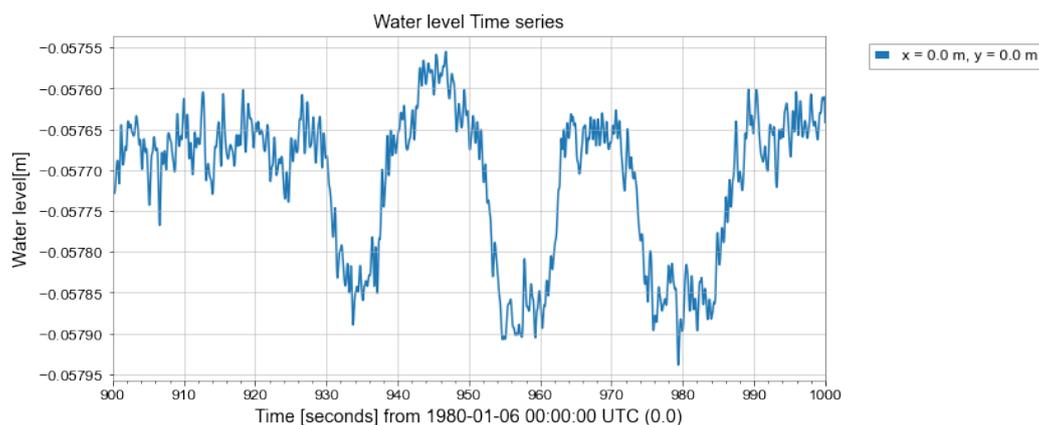


図 5.4: 水位の時系列の例。初期流量 109 t/h の条件の  $(x, y) = (0, 0)$  地点。乱流のため、状態が収束しても水面が一定値に落ち着くことはない。

くことが分かる。これはパイプの排水能力が流入量を上回っていると見るができる。KAGRA のパイプは中央排

水管で排水できない量の水が流れる場合は1~3本の強制排水管を併用して流すため、常にパイプの排水能力が流入量を上回るこの状態になると予想できる。

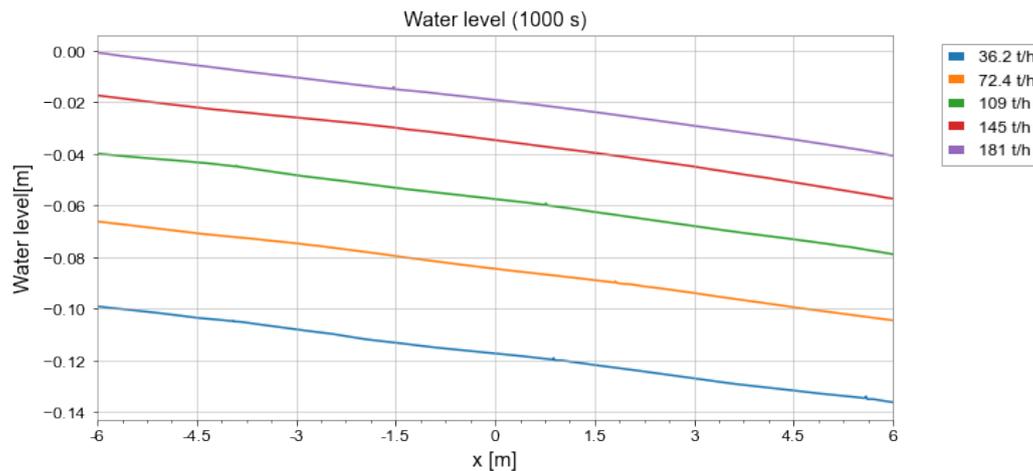


図 5.5: 各条件における  $y = 0$  m 地点で  $x$  方向への水位。シミュレーション終了時間の 1000 秒時。どの条件においても下流方向に水位が低くなっていくことが分かる。また、初期流量が少ないほど、水位も低くなっていることが分かる。

次に水面のゆらぎを比較するためにパイプの中心 ( $y = 0$  m) の線上で  $x = -6$  m から  $x = 6$  m までの範囲で等間隔に地点を選び、その地点での水面のゆらぎを横軸周波数、縦軸をゆらぎの強さにして比較した (図 5.6)。この範囲は現実の KAGRA のパイプではテストマスは入口と出口から遠いため、本シミュレーション独自の入口と出口の影響を除くためである。さらに全体のゆらぎの傾向を見るためにパイプの中心線に沿った地点の  $x = -6$  m ~  $6$  m の範囲の水面のゆらぎを横軸  $x$  方向、縦軸周波数、色をゆらぎの強さにした図 5.7 も作成した。まず、ゆらぎの水面のゆらぎのスペクトル図 5.6 を見る。どの条件にもおいてもほとんどの地点で 2 ~ 3 Hz から高周波側にかけて急激にゆらぎが小さくなっていることが分かる。さらに、水面のゆらぎを周波数で見ると各水面で前後の地点と似たような概形をとっていることが分かる。よって特別な外的要因がなく、同じ水面上であればどの地点でも似たような周波数のゆらぎを持つと言える。しかし、36.2 t/h や 109 t/h において幾つかの地点で他の地点と明らかに違うゆらぎの概形をしていることが分かる。これらの地点は高周波側で高いゆらぎを持っており、重力勾配雑音が水面のゆらぎと直接的な関係にある性質からも無視できない地点である。特に KAGRA は 100 Hz 付近の重力波検出を目指しており、この地点で 100 Hz 付近のゆらぎが大きいか調べる必要がある。この地点を詳しく調べるため、等間隔でいくつかの場所を取るのではなく  $y = 0$  m ( $-6$  m  $< x < 6$  m) 上全ての計測地点のゆらぎを図 5.7 に示した。

図 5.7 について詳しく見る。図 5.7 を見ると、色が黄色に近いほどゆらぎが多きく青いほどゆらぎが小さいことを示すが、どの条件においても全ての地点で低周波側から 1 Hz より少し大きい 2 ~ 3 Hz あたりまでゆらぎが大きいことが分かる (図 5.6 においても同様の傾向が見られる)。このことから直線状のパイプの水面の大部分では、低周波側から 2 ~ 3 Hz までのゆらぎが発生しており、重力勾配雑音ではこの周波数までの影響が大きいことが考えられる。特に、181 t/h の条件では多くの地点で他の条件に比べてゆらぎの大きさが小さい。しかし、181 t/h の条件で他の条件に比べて、全体的にゆらぎが小さい原因は判明しなかった。(仮説としては、水量が多い、または径深が深い場合は凹凸のある壁面から水面までの距離が遠く、凹凸の影響を受けにくいのでゆらぎが小さいと考えられる。) さらに、145 t/h と 181 t/h の条件では上流側から 2 ~ 3 Hz の周波数部分が特に濃くなっている。これは何らかの原因で 2 ~ 3 Hz がピークとなるゆらぎが水面で発生していることを示す。

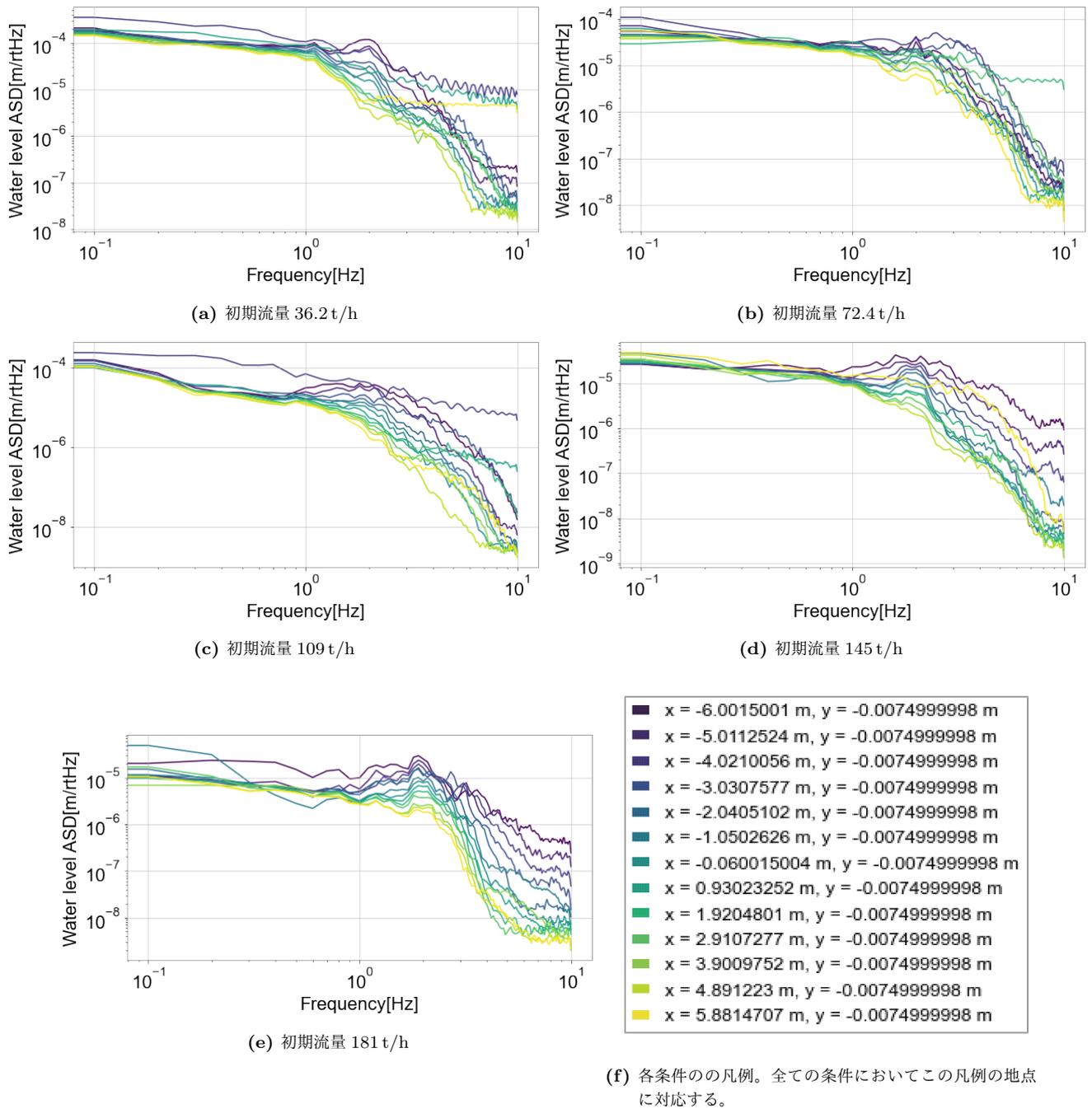


図 5.6: 各条件において、パイプ中心 ( $y = 0$  m) の線上で  $x = -6$  m から  $x = 6$  m の地点まで等間隔の地点における水面のゆらぎの比較。地点は凡例の通りで、色が紫に近い方が上流に近く、黄色に近いものが下流に近い。横軸は周波数で縦軸はゆらぎの大きさとなっている。いくつかの地点で高周波側においてもゆらぎが大きい地点が存在している。

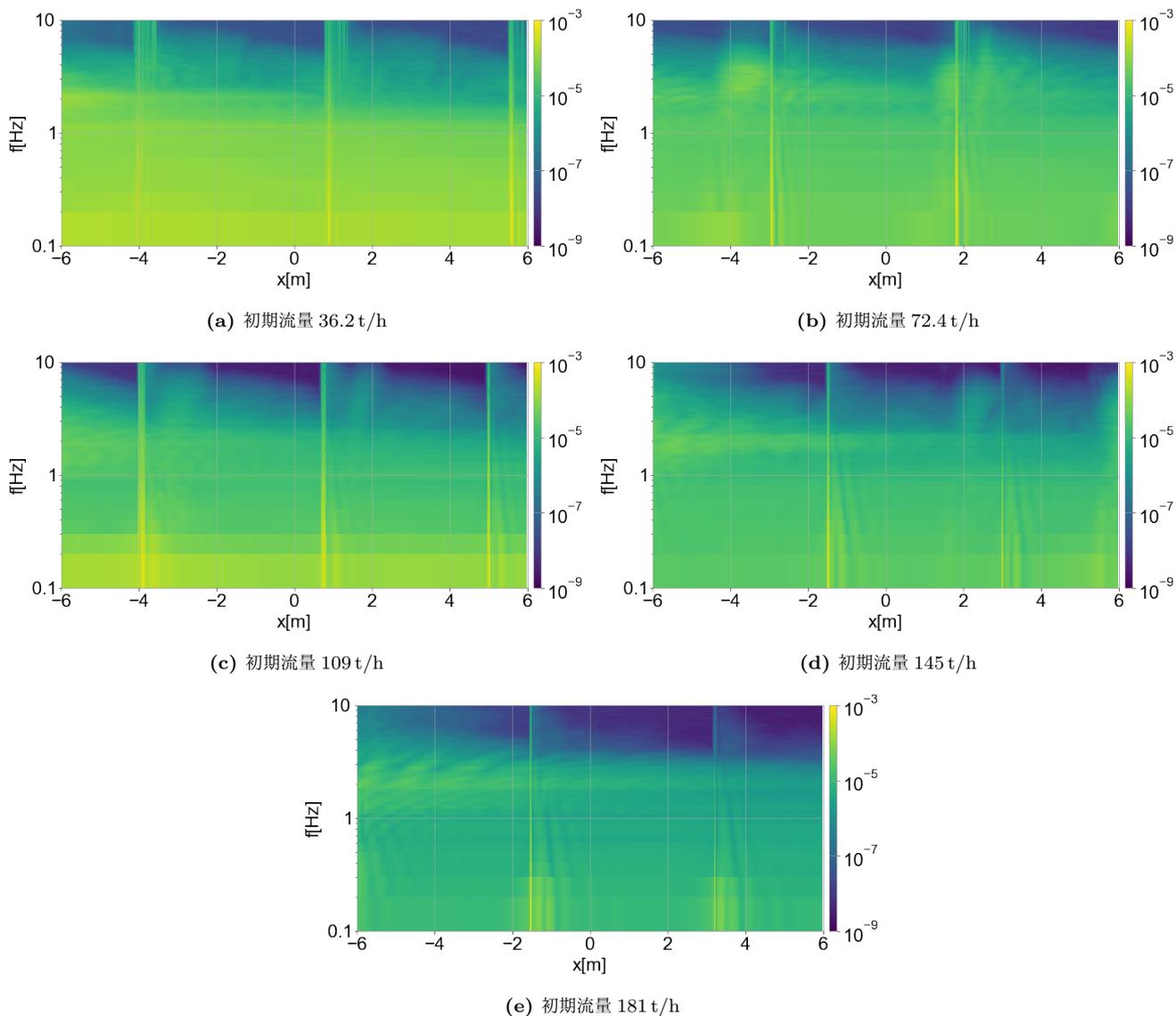


図 5.7: 各条件において、パイプ中心 ( $y = 0$  m) の線上で  $x = -6$  m から  $x = 6$  m の地点までの水面のゆらぎの比較。縦は周波数、横は  $x$  方向の座標、色はゆらぎの強度を表す。全ての条件で水面の大部分で低周波側から 2 ~ 3 Hz までのゆらぎが大きいことが分かる。また、黄色い縦線の部分は局所的に高周波側までゆらぎが大きい地点である。

ここまでで図 5.7 では水面の多くの場所では 2 ~ 3 Hz までのゆらぎが大きいことが分かり、さらに図 5.6 を見た時にも指摘した、局所的に高周波側で高いゆらぎを持つ地点を図 5.7 で見る事ができた。その地点は全て条件で見ることができ、0.1 Hz ~ 10 Hz で黄色い線上になっている地点である。黄色い線上になっているということは見えている周波数 0.1 Hz ~ 10 Hz でゆらぎが大きいということであり、さらに高周波においてもゆらぎが大きい可能性があるということである。図 5.7 を見ることで、全ての条件でこの高周波側に大きいゆらぎを持つ地点が発生することが分かった。またこの地点は一つの水路で 2、3 箇所発生していることも分かる。

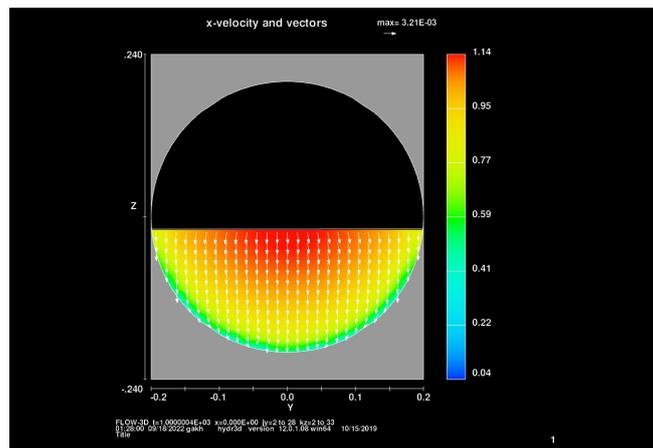
局所的揺らぎの発生地点は限定的であり、重力勾配雑音への影響も 2 ~ 3 Hz 以下の周波数に比べれば小さくなることが予想できる。しかし、重力勾配雑音はテストマスとの距離が近い場所の影響を大きく受けるのでこの線が発生する領域が狭いものであってもテストマスと最も近い  $x = 0$  m 地点に近い場所で発生する場合には、重力勾配雑音に大き

く影響を与える可能性がある。そのため、前述したように KAGRA は 100 Hz 付近の重力波検出を目指しているので、100 Hz で大きなゆらぎを起こす可能性があるこの地点の発生原因の考察とどの程度のゆらぎが発生するかを確認することが KAGRA にとって重要となる。局所的揺らぎについては、直線状のパイプの多くの条件で見ることができるため、この章の後半にて詳しく考察する。

### 5.3.2 パイプの断面・ $x$ 方向の流速

流速についての情報を参考として紹介する。流速については水面のゆれとは直接的に関係がなく、この情報から重力勾配雑音への影響について考察することは困難であるが、パイプ内の水の状態を知る情報として有用である。ここでは Flow-3D から出力されるパイプ断面の流速 (181 t/h) と、水面直下の  $y = 0$  m の  $x$  方向の流速を示す (図 5.8、図 5.9)。前者は流速の分布が分かると同時に、パイプ断面のどの位置まで水が存在するのかが分かる。後者は水の進み具合を見ることができ、水の流れをイメージしやすいものとなっている。

図 5.8 を見るとパイプ壁面付近では流速が遅く、壁面から離れるほど流速が速いことが分かる (パイプ中心線の水面付近が最も速い)。流速が変化している部分では周りの水の移動速度と差ができるということであり、その差が波となって現れる可能性がある。



(a) 初期流量 181 t/h

図 5.8: 初期流量 181 t/h においてパイプの真ん中 ( $x = 0$  m) のパイプ断面の  $x$  方向の流速の分布。色が赤いほど流速が速い。色がついている部分に流体があるので、流体がどこまで入っているかが分かる。流速はパイプ中央の水面付近で最も早く、パイプ壁面付近で遅くなっていることが分かる。また、他の条件においても同様の速度の分布が見られた。

図 5.9 を見ると、基本的に流速は出口に向かって速くなっていることが分かる。また、全ての条件で出口付近で流速が速いことが分かるが入口の様子は条件によって異なっていることが分かる。初期流量 36.2 t/h、72.4 t/h の条件では入口で流速が速く、そこから急激に遅くなっているが他の条件では入口の流速が遅く、そこから緩やかに増加している。考察として本シミュレーションでは入口から流入する時の高さをどの条件も同じにしているが、流量が少ない条件では入口以外の水面の高さが低いいため入口で急激に水面の高さが下がることが原因であると考えられる。また、出口付近で流速が速いことはベルヌーイの定理から説明でき、出口付近では圧力が低下するため、流速が速くなる。

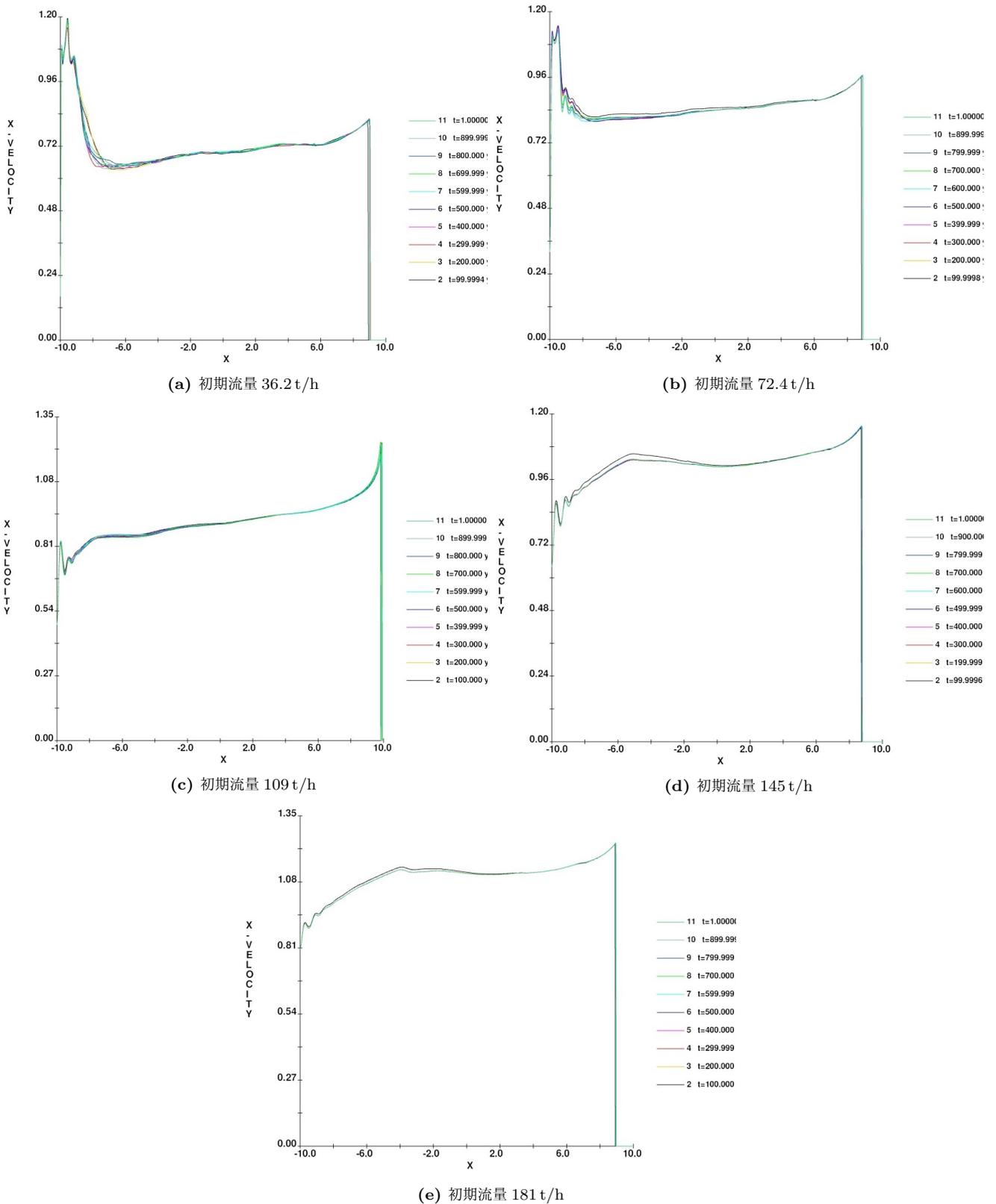


図 5.9: 各条件においてパイプの中心線 ( $y = 0$  m) で、水面に近い位置の  $x$  方向の流速の大きさの  $x$  方向の分布。縦軸は  $x$  方向の流速の大きさ、横軸は  $x$  方向の位置となっている。流量の多い条件と少ない条件で入口の様子が異なっているが、これは水の流入高さが一定であり流量の少ない条件では水面との差が生じているためだと考えられる。また、基本的に流速は出口に向かって速くなっている。

## 5.4 相当粗度 (壁面粗さ) の違いによるシミュレーションの比較の解析結果

### 5.4.1 水面のゆらぎ

初期流量の比較と同様に水面のゆらぎから考察していく。基本的には初期流量で行った考察の流れに沿うように記述する。シミュレーションにかかった時間とデータサイズを表 5.3 にまとめた。これを見ると相当粗度 0.1 mm の条件だけ他の条件より少し時間がかかっている。これは相当粗度の値が小さいほど壁の抵抗が少なくなり、流速が速くなることで計算に時間がかかるためだと考えられる。流速については詳しく後述する。ただしシミュレーションに数日かかる点は同様である。

表 5.3: 直線状のパイプで相当粗度を比較したシミュレーションのシミュレーションにかかった時間やシミュレーションのデータサイズ

番号	相当粗度 [mm]	シミュレーション時間	シミュレーションにかかった時間 [s]	データサイズ [GB]
S16	0.1	1010 秒	826496	182
S17	0.4	1010 秒	633502	183
S18	1.2	1010 秒	555235	183
S19	2.0	1010 秒	619151	183
S1	3.2	1010 秒	548207	184

図 5.10、では水位の比較を行い、図 5.11 では初期流量の比較と同様に  $x$  方向に等間隔の地点をとり、ゆらぎの比較を行った。図 5.10 を見ると初期流量の比較で見た時と同様に上流側の水位が高く、下流側の水位が低くなっており、本シミュレーションの条件ではどの相当粗度であってもパイプ内の水が正常に流れていることが分かる。また、相当粗度の値が小さいほど水位が低いことが分かる。これは相当粗度の値が小さいほど水がパイプ壁面から受ける抵抗が少なく、パイプの排水能力が高いためだと考えられる。

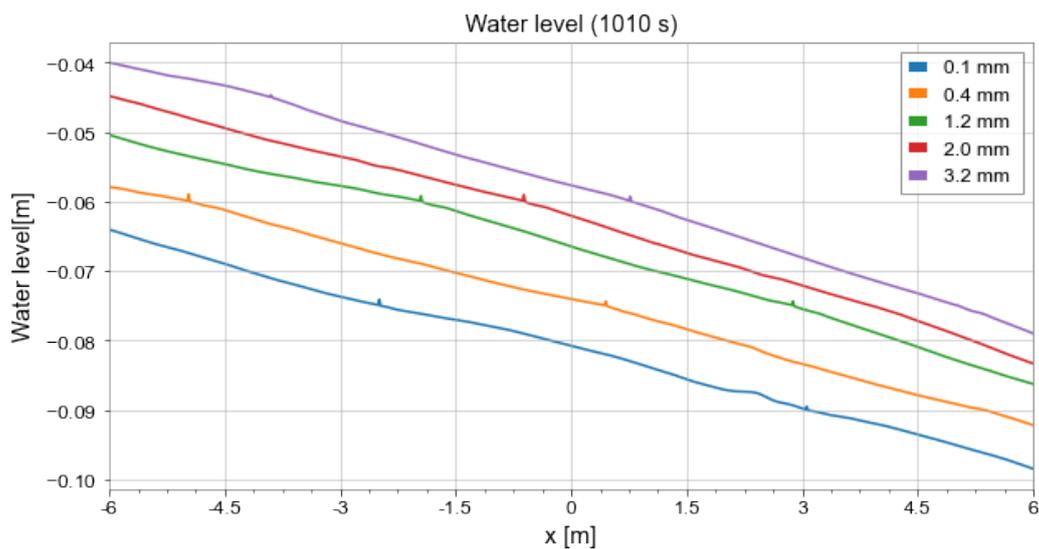


図 5.10: 各条件における  $y = 0$  m 地点で  $x$  方向への水位。シミュレーション終了時間の 1010 秒時。どの条件においても下流方向に水位が低くなっていくことが分かる。

図 5.11 を見ると初期流量の比較においても見られた局所的に高周波側にかけてゆらぎが大きい場所が存在する。そのため初期流量の比較の時と同様に図 5.12 のように水面のゆらぎを詳しく見た。図 5.12 を見ると、初期流量の比較の場合と同様に水面の多くの地点で 2 ~ 3 Hz あたりまでのゆらぎが大きくそれより高周波側ではゆらぎが小さくなっていることが分かる。よって相当粗度の比較においても重力勾配雑音は低周波側から 2 ~ 3 Hz までの周波数で雑音が大きいたことが考えられる。また、相当粗度の比較においても黄色い線となっている局所的に揺らぎが大きい地点が確認できる。ただし、相当粗度 0.4 mm、1.2 mm、2.0 mm の条件ではこの地点のゆらぎが小さく、図 5.12 を見るとかなり限定的なゆらぎとなっている。さらに相当粗度 0.1 mm の条件で上流側の地点で 2 ~ 3 Hz の周波数のゆらぎが大きいたことが分かる。これは初期流量で見た時と同様に入口に近い地点におけるもので、入口の影響が考えられる。初期流量では流量が多い条件で見られ、相当粗度では小さい条件で見られたことが特徴である。

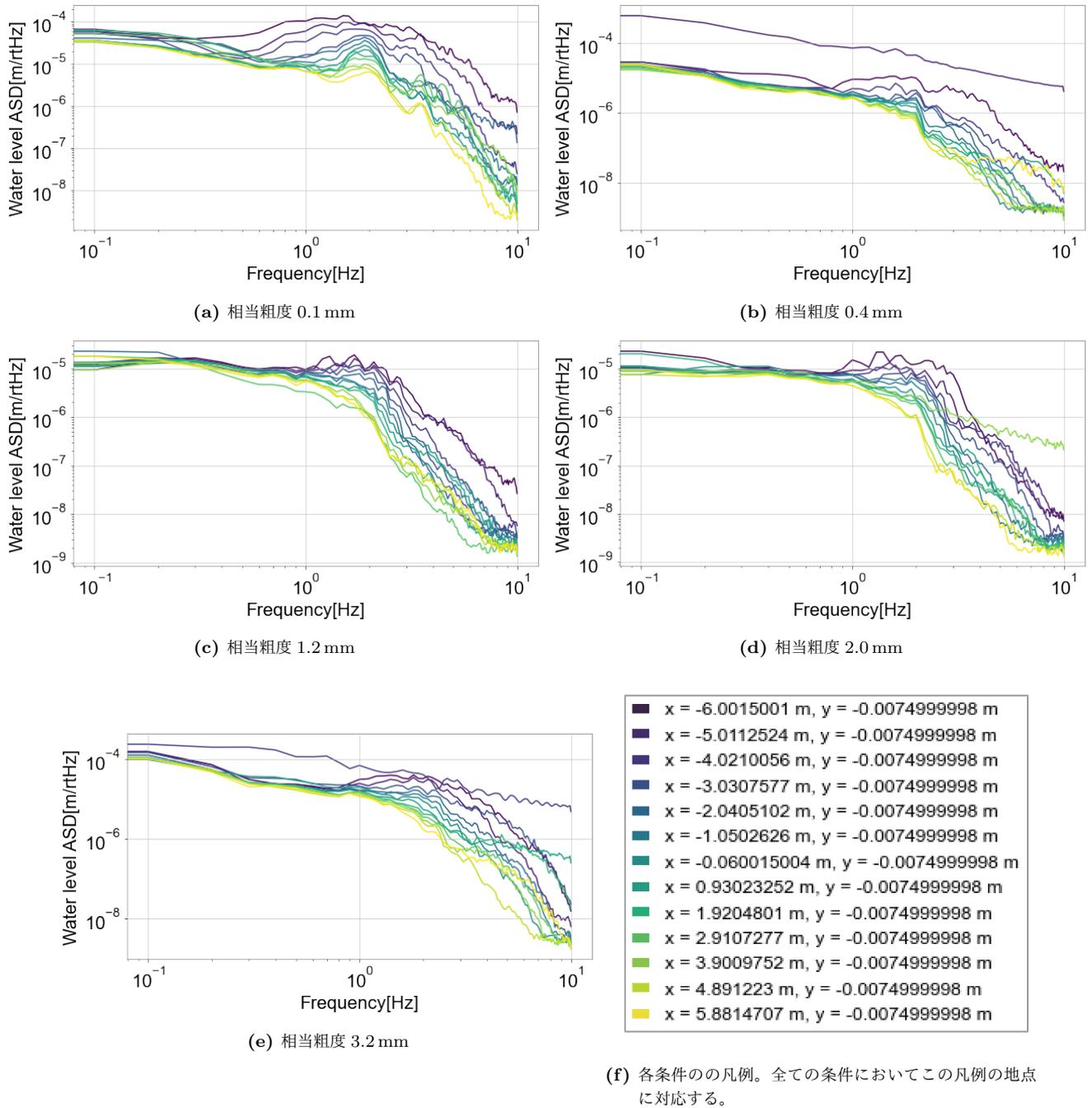


図 5.11: 各条件において、パイプ中心 ( $y = 0$  m) の線上で  $x = -6$  m から  $x = 6$  m の地点まで等間隔の地点における水面のゆらぎの比較。地点は凡例の通りで、色が紫に近い方が上流に近く、黄色に近いものが下流に近い。横軸は周波数で縦軸はゆらぎの大きさとなっている。

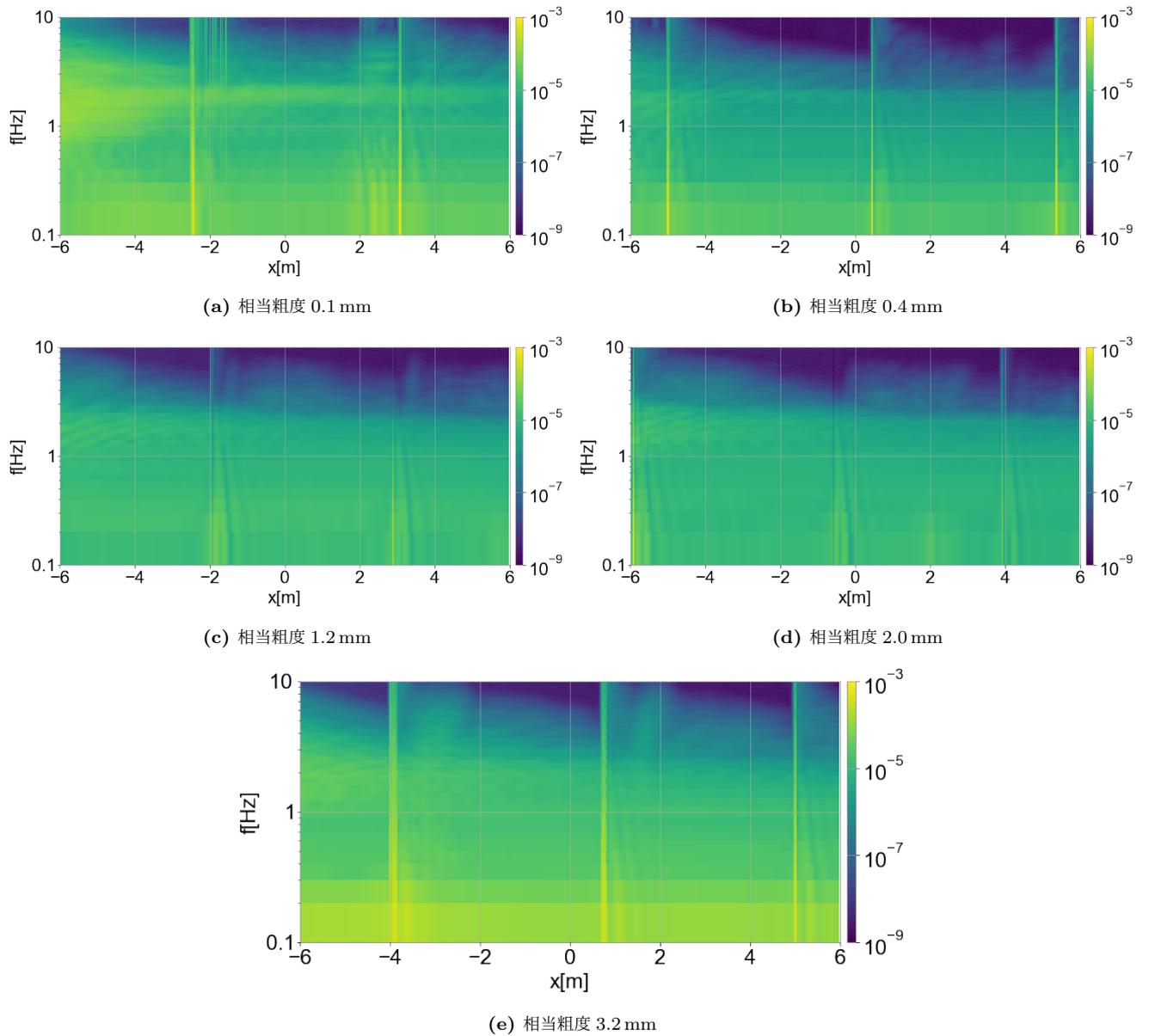


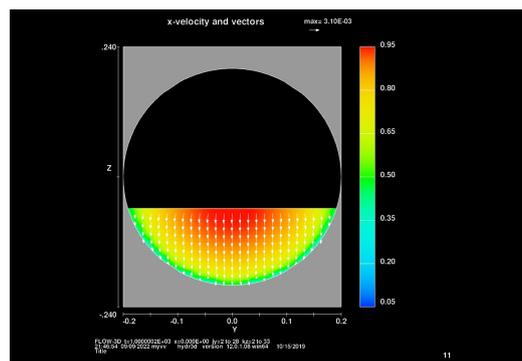
図 5.12: 各条件において、パイプ中心 ( $y = 0$  m) の線上で  $x = -6$  m から  $x = 6$  m の地点までの水面のゆらぎの比較。縦は周波数、横は  $x$  方向の座標、色はゆらぎの強度を表す。

### 5.4.2 パイプの断面・ $x$ 方向の流速

初期流量の比較の時と同様に流速についての情報を参考として紹介する。ここでは Flow-3D から出力されるパイプ断面の流速 (相当粗度 3.2mm) と、水面直下の  $y = 0$  m の  $x$  方向の流速を示す (図 5.13、図 5.14)。前者は流速の分布が分かると同時に、パイプ断面のどの位置まで水が存在するのかが分かる。後者は水の進み具合を見ることができ、水の流れをイメージしやすいものとなっている。

図 5.13 を見ると初期流量 181 t/h の条件の断面の速度分布で見られたものと同等な速度分布が見られる。また、他の相当粗度の条件においても同様の速度分布が確認できた。これはパイプの形 (内径の大きさや直線状であること) が同等であれば流速の分布は変わらないと考えられる。また、パイプ壁面付近では流速が遅く、壁面から離れるほど流速が速いことが分かる (パイプ中心線の水面付近が最も速い)。流速が変化している部分では周りの水の移動速度と差ができることであり、その差が波となって現れる可能性がある。

図 5.14 を見ると、全ての条件で出口付近で急激に流速が速くなっていることが分かる。これは初期流量の比較においても言及したが、ベルヌーイの定理から説明できる現象である。また、相当粗度が小さいほど入口から出口まで流速の変化が少ないことが分かる。考察として、抵抗が少ないほうがどの地点でも水が速度を保ったまま流れるのではないかと考えられる。さらに、入口付近に注目すると全ての相当粗度の条件で同じ様な挙動であることが分かる。これは相当粗度の比較では全ての条件で初期流量 109 t/h であり、水の流入高さと水面の関係がほぼ同じであるからだと考えられる。



(a) 相当粗度 3.2 mm

図 5.13: 相当粗度 3.2mm において、パイプの真ん中 ( $x = 0$  m) のパイプ断面の  $x$  方向の流速の分布。色が赤いほど流速が速い。色がついている部分に流体があるので、流体がどこまで入っているかが分かる。また、他の粗度係数の条件においても同様の速度の分布が見られる。

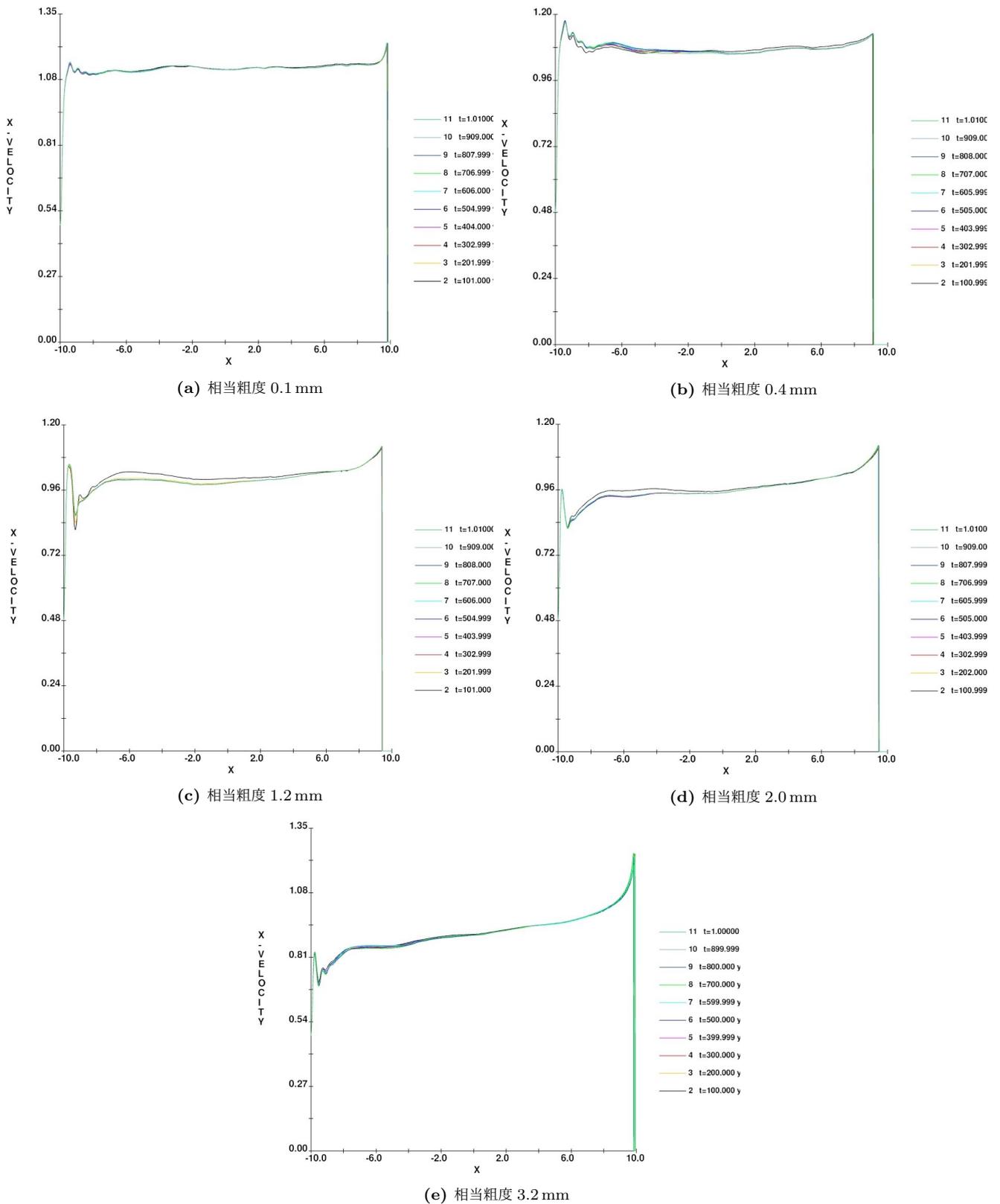


図 5.14: 各条件においてパイプの中心線 ( $y = 0\text{ m}$ ) で、水面に近い位置の  $x$  方向の流速の大きさの  $x$  方向の分布。縦軸は  $x$  方向の流速の大きさ、横軸は  $x$  方向の位置となっている。出口では急速に流速が速くなっている。相当粗度の値が小さくなるほど、 $x$  方向への流速の変化が少なくなる。

## 5.5 傾斜の違いによるシミュレーションの比較の解析結果

シミュレーションにかかった時間をまとめたものを表 5.4 にまとめた。傾斜の違いを調べたシミュレーションでは、シミュレーション時間で大きな差が出た。傾斜 0.3 % の条件の方が 0.13 % の条件よりもシミュレーション時間が長かった。これは傾斜 0.3 % の方が流速が速く、計算に時間がかかるためだと考えられる。さらに、 $x$  方向への水位の分布 (図 5.15) を見るとこれまでの直線状のパイプと同様に出口に向かうにつれて水位が下がっていることが分かる。また、傾斜がゆるやかな 0.13 % のほうが水位が高い。これは 0.13 % のほうがパイプの排水能力が低いことを示している。

表 5.4: 直線状のパイプで傾きを比較したシミュレーションのシミュレーションにかかった時間やシミュレーションのデータサイズ

番号	傾斜 [%]	シミュレーション時間	シミュレーションにかかった時間 [s]	データサイズ [GB]
S20	0.13	1010 秒	290454	183
S1	0.3	1010 秒	548207	184

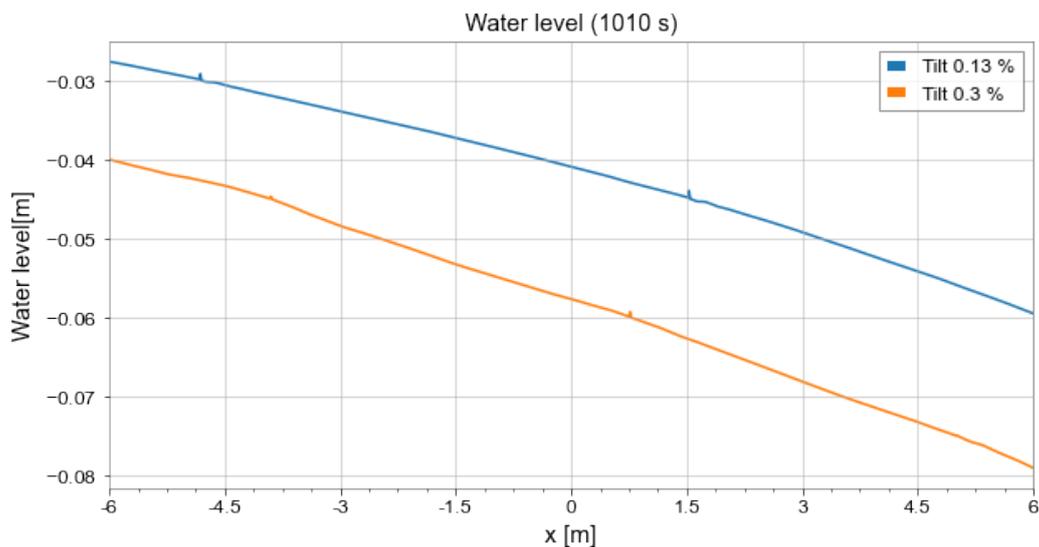


図 5.15: 各条件における  $y = 0$  m 地点で  $x$  方向への水位。シミュレーション終了時間の 1010 秒時。どちらの傾斜においても下流方向に水位が低くなっていくことが分かる。

傾斜の違いによるシミュレーションの比較においても水面のゆらぎを比較した。水面のゆらぎを比較した (図 5.16) を見ると水面の様子が異なっていることが分かる。初期流量の比較と相当粗度の比較でも見えたゆらぎが局所的にゆらぎが大きい地点の場所や個数も変化していることが分かる。よってこの地点は傾斜による違いでも現れ方に変化を起し、パイプに関する条件が少しでも変れば変わるため制御が難しいと言える。なお重力勾配雑音を考えると、局所的に大きなゆらぎがある地点は限定的でありこの地点以外の水面のゆらぎの状況にあまり変化がないので大きなゆらぎが影響をもたらす 10 Hz 以上の部分を除いた低周波側の重力勾配雑音は両条件であまり変化がないことが予想される。

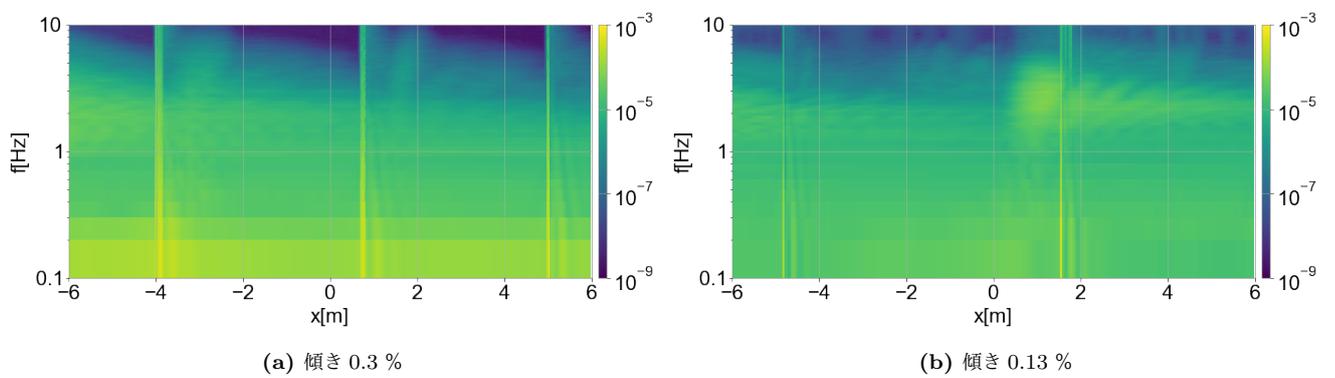


図 5.16: 各条件において、パイプ中心 ( $y = 0$  m) の線上で  $x = -6$  m から  $x = 6$  m の地点までの水面のゆらぎの比較。縦は周波数、横は  $x$  方向の座標、色はゆらぎの強度を表す。

### 5.5.1 パイプの断面・ $x$ 方向の流速

速度分布 (図 5.17) を見ると、速度面では傾斜が大きいほうが速いが、分布の仕方などはほぼ同様であることが分かる。また、 $x$  方向への流速の分布についても同様の分布になっている。よって速度分布は 0.3 % と 0.13 % の傾斜の違いでは大きく変化しないことが分かる。

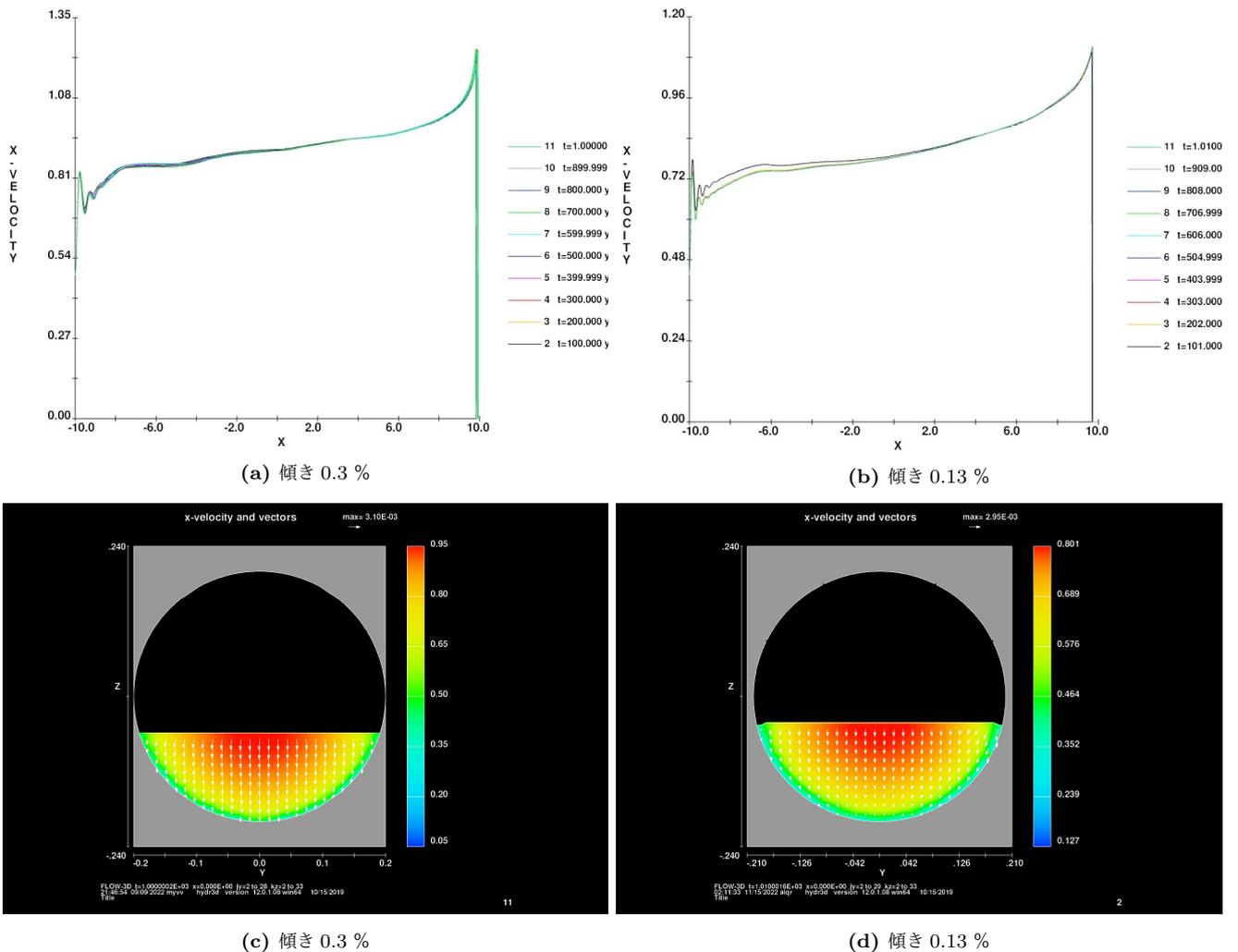


図 5.17: パイプの中心線 ( $y = 0$  m) で、水面に近い位置の  $x$  方向の流速の大きさの  $x$  方向の分布 (上段)。パイプの真ん中 ( $x = 0$  m) のパイプ断面の  $x$  方向の流速の分布 (下段)

## 5.6 局所的に大きなゆらぎを持つ地点

直線状のパイプにおいて多くの条件で確認された局所的に高周波側まで大きなゆらぎを持つ地点について考察する。このゆらぎは10 Hzよりも高周波においてもゆらぎが大きい可能性があり、その場合10 Hz以上の水の重力勾配雑音を下げるためには回避しなければならない重要な地点となる。なお、このゆらぎは局所的なため、重力勾配雑音への影響は少ないと考えられるが、どの程度重力勾配雑音に影響に与えるかについては7章にて考察する。なお、これより局所的に大きなゆらぎを持つ地点を「特異変動水面」と呼称する。

### 5.6.1 初期流量との関係

ここでは局所的に大きなゆらぎを持つ地点(図5.7で黄色い部分を持つ地点)についての考察を行う。見える周波数で他の地点より大きいゆらぎを持つということは、単純に他の地点より水面の動きが激しいと考えられる。まず、注目する地点が10 Hz以上でゆらぎが大きいことを確かめるために高周波帯(10 Hz~100 Hz)の水面のゆらぎを見る(図5.18)。このデータは1000秒から1010秒のものを使用している。取得している秒数が少ないため、断続的な周期を持つゆらぎなどはうまく捉えられていない可能性があるが、参考にした。図5.18を見ると特異変動水面では100 Hzまでゆらぎが大きいことが分かる(100 Hzまで黄色い線となっている)。10 Hz以上のゆらぎは特異変動水面のみで大きく、10 Hz以上の周波数における重力勾配雑音の大きさは特異変動水面の影響を受けることが考えられる。よってKAGRAを意識した場合、重力勾配雑音の影響を少なくするためにはこの大きなゆらぎを持つ地点の影響を最小限にとどめることが最重要になると考えられる。さらに、図5.18を見ると特異変動水面より下流側の0.1 Hz~10 Hz付近では特異変動水面の部分からゆらぎが伝搬しているように見える。なお、10 Hz以上のゆらぎは下流側には伝搬していないことが分かる。

次に3章で紹介した水理学の視点から考えると、定常流の条件で、単純な直線水路において水の動きが激しくなる地点は「跳水」と推測できる。他にも、パイプの断面形状に起因するものや、図5.8で見たようにパイプ中央と壁面付近の速度の違いによって引き起こされている可能性も考えられる。3章で紹介したように跳水の発生位置は射流から常流となる場所であった。つまり、特異変動水面が跳水であれば、上流側が射流、下流側が常流となる。また、跳水(または、常流、射流)であるかはフルード数により確認することができ、Flow-3Dの出力機能を用いて確認を行った。なお、初期流量の比較では初期流量と特異変動水面の個数や地点に相関が見られず、初期流量と特異変動水面の個数や場所に関係性を見出すことはできなかった。

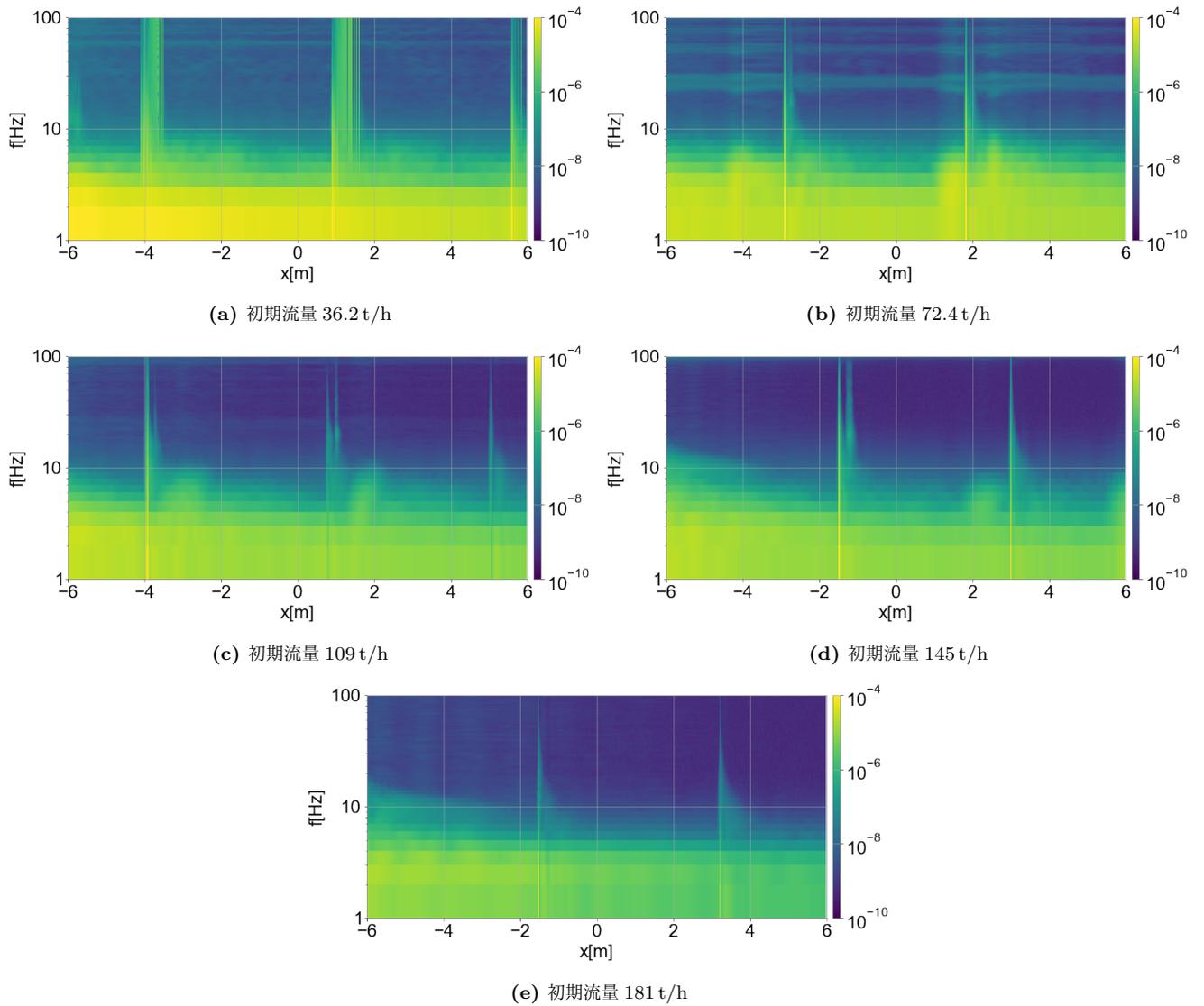


図 5.18: 各条件において、パイプ中心 ( $y = 0\text{ m}$ ) の線上で  $x = -6\text{ m}$  から  $x = 6\text{ m}$  の地点までの水面のゆらぎの比較。縦は周波数、横は  $x$  方向の座標、色はゆらぎの強度を表す。1000 秒から 1010 秒の水面のデータを使用している。特異変動水面では 100 Hz までゆらぎが大きいことが分かる。特異変動水面では明らかに他の地点とは異なるゆらぎの性質を示している。重力勾配雑音が水面のゆらぎから影響を受けることを考えると、重力勾配雑音の 10 Hz 以上の大きさは特異変動水面の影響の変化と関わっていると考えられる。

### 跳水であるかの考察

特異変動水面が跳水であるかを調べるために Flow-3D 機能を使用し、パイプの中心線のフルード数を出力した (図 5.19)。この値は  $z$  座標に依らない値である。3.1.2 節でも紹介したようにフルード数は流速と小重力波の伝搬速度の比であり 1 より大きいと射流、小さいと常流である。そしてこの射流から常流になる境界において急激に水位が上がる場所が跳水となるので、跳水となる場所はフルード数が 1 以上から 1 以下になる場所となる。なお、跳水は水位が上昇している場所なので、跳水の波が発生している地点のフルード数は 1 以下の常流の状態になっていると考察できる。ただし、フルード数は理論値であり本シミュレーションでは断面が円形のパイプを使用していることや平均流速の取り方によってフルード数が増えるので、跳水が発生するフルード数が 1 になっていない可能性も考えられる。

図 5.19 を詳しく見る。出口に向かってフルード数が大きくなっているのは出口に向かって水深が浅く (図 5.5)、流速が速くなっている (図 5.9) からだと考えられる。全ての条件で、フルード数が 0.7 前後の領域が多いことが分かる。また、36.2t/h と 72.4t/h の条件では入口と出口を除けばフルード数がほぼ一定であるが 109t/h 以上の条件では -6m 付近で一度フルード数が大きくなり下流に向かうと小さくなるという山が見られる。このシミュレーション条件において跳水が起こる基準となるフルード数が 0.7 付近にあった場合この山の下流側のフルード数が下がる (射流から常流になる) 領域で跳水が発生する可能性がある。

ここで一回結論を出すと、36.2t/h と 72.4t/h はフルード数の変化がほぼなく、跳水が起こっている可能性は低い。しかし、図 5.18 を見ると特異変動水面が存在しており、この条件の特異変動水面は跳水以外の原因で起こっている可能性を考えるべきである。ただし、このパイプの構造で跳水が起こる基準がフルード数 0.7 付近である場合には跳水が起こり易い状況であると言える。一方 109t/h 以上の条件ではフルード数が下がっている領域があり、跳水が起こる基準となるフルード数が 0.7 付近だった場合この領域で跳水が起こる可能性があり図 5.18 で見える上流側 (フルード数の山がある部分) の特異変動水面は跳水である可能性を指摘できる。しかし下流側では射流に向かっており、跳水は発生しないので、下流側の特異変動水面は跳水ではないと考えられる。仮説としてこれら跳水ではない特異変動水面は、パイプの構造が原因となる場合や、流速が急激に上昇したことなどが原因だと考えられる。また、フルード数を見ると初期流量の比較では全ての条件で常流の状態にあることが分かる。よって初期流量の比較における重力勾配雑音は常流の状態の水流から受ける重力勾配雑音と言える。なお、フルード数が 1 を越える状態のゆらぎを観察できた場合、ここで見たような水面のゆらぎとは違う構造が現れ、重力勾配雑音においても異なる特徴を示す可能性がある。

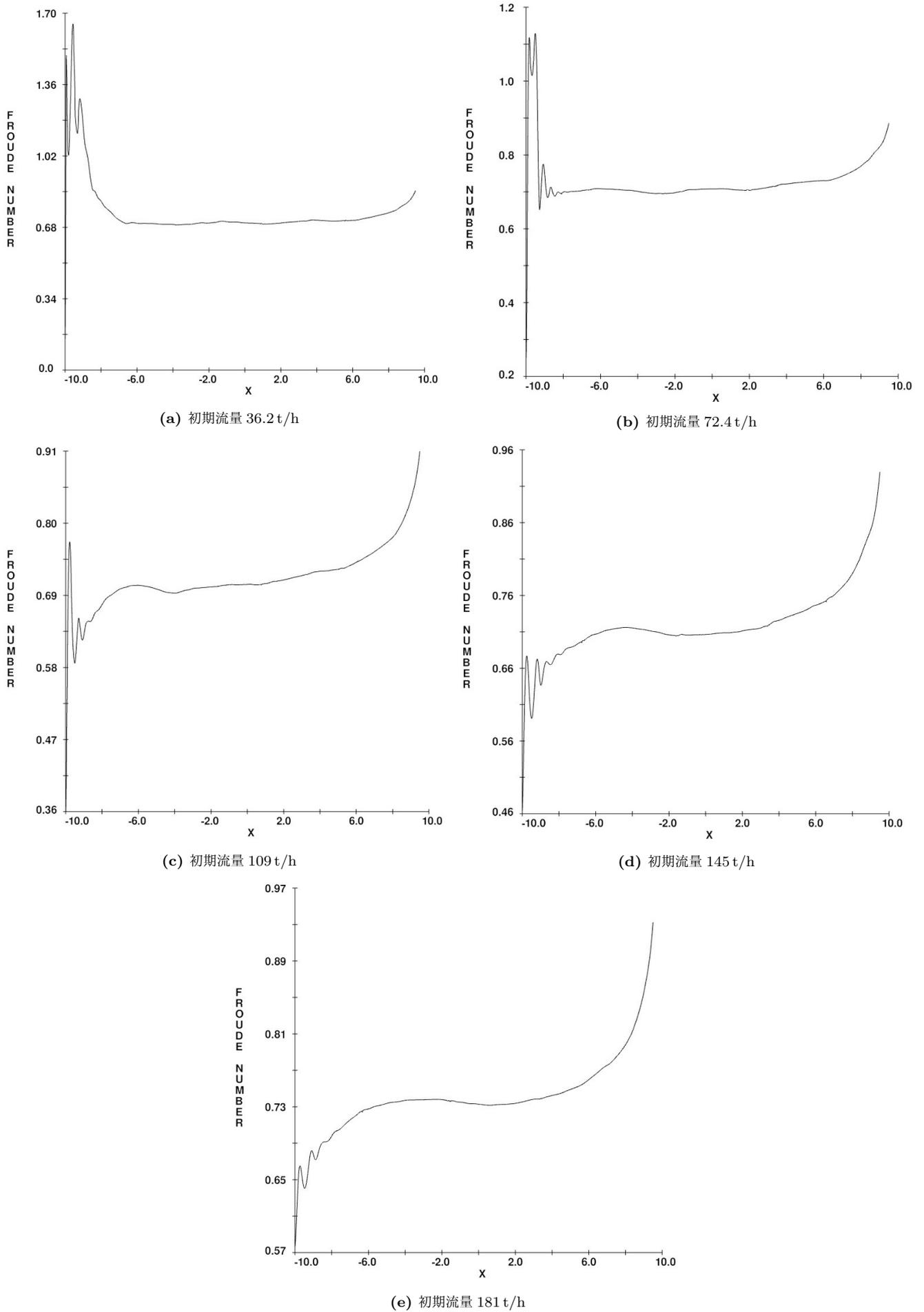


図 5.19: 各条件において、パイプ中心 ( $y = 0\text{ m}$ ) の線上のフルード数の  $x$  方向への分布。全ての条件でフルード数は 0.7 付近であり、このフルード数を信頼するならば全ての条件において常流の状態であると言える。なお跳水が発生している場合はフルード数が下がるので 109 t/h 以上の条件で見えるフルード数が下がっている場所で跳水が発生している可能性がある。

## 5.6.2 相当粗度との関係

図 5.12 を見ると、相当粗度の比較においても特異変動水面が複数地点に現れていることが分かる。相当粗度の比較においても高周波帯の水面のゆらぎ図 5.20 を確認した。これを見ると、特異変動水面のゆらぎは初期流量の比較で見た時と同様に 100 Hz までのゆらぎが大きいことが分かる。よって重力勾配雑音を小さくするためには、相当粗度を変化させて発生した特異変動水面もなるべくテストマスから距離を離すなどして影響を小さくする必要がある。また、相当粗度 0.4 mm、1.2 mm、2.0 mm の条件では特異変動水面の 10 Hz 以上のゆらぎが他の条件と比べて小さい。そのため、この 3 条件の重力勾配雑音の 10 Hz 以上のゆらぎは相当粗度 0.1 mm、3.2 mm の条件と比べて小さい可能性がある。なお、相当粗度の比較においても相当粗度と跳水と思われる地点の個数や場所に関係性を見出すことはできなかった。次に初期流量の比較で行ったように相当粗度の比較で発生した特異変動水面が跳水であるかを Flow-3D より出力できるフルード数によって確認する。

### 跳水であるかの考察

初期流量の比較の時と同様に跳水であるかどうかの考察を行う。同様に  $x$  方向のフルード数の分布について Flow-3D から出力した (図 5.21)。これを見ると初期流量の比較で見たように出口付近ではフルード数が上がっている。さらに全体的なフルード数は相当粗度が小さくなるほどに上がっていることが分かる。これは相当粗度が小さいほうが流速が速くなっている (図 5.14) ためだと考えられる。また、相当粗度 1.2 mm 以上の条件では初期流量の比較で見た時と同様に上流側にフルード数の山ができていたため、この山付近のフルード数が跳水が発生する基準となっていた場合跳水が発生する。ただし図 5.20 を見ると、相当粗度 1.2 mm と 2.0 mm の条件では特異変動水面のゆらぎが小さいため、大きなゆらぎとなる跳水が発生していないと考えられる。よってここでは跳水が発生するフルード数は 0.7 付近ではないと言える。また、相当粗度の比較のシミュレーションにおいても 0.1 mm の条件以外ではフルード数は 1 よりも小さく、常流の状態にあると考えてよい。よって相当粗度の比較における重力勾配雑音も常流の状態の水面から受ける重力勾配雑音と言える。

ここで相当粗度 0.1 mm の条件に注目する。この条件では大部分の領域でフルード数が 1 に近い。そのため理論通りにフルード数が 1 で跳水が起こるとすれば非常に跳水が起きやすい状態であると言える。よって相当粗度 0.1 mm の条件の特異変動水面は跳水である可能性が高い。また、この相当粗度は実際に測定されたパイプの値から計算された相当粗度に近い条件であり、実際の KAGRA のパイプにおいてもフルード数が 1 に近く跳水が発生しやすい可能性が考えられる。一方、3.2 mm の条件の特異変動水面はフルード数を信頼すると跳水でない別の原因で発生していると考えられる。ただし、-4 地点の特異変動水面はフルード数が低下する地点であり跳水に似た現象が発生している可能性がある。

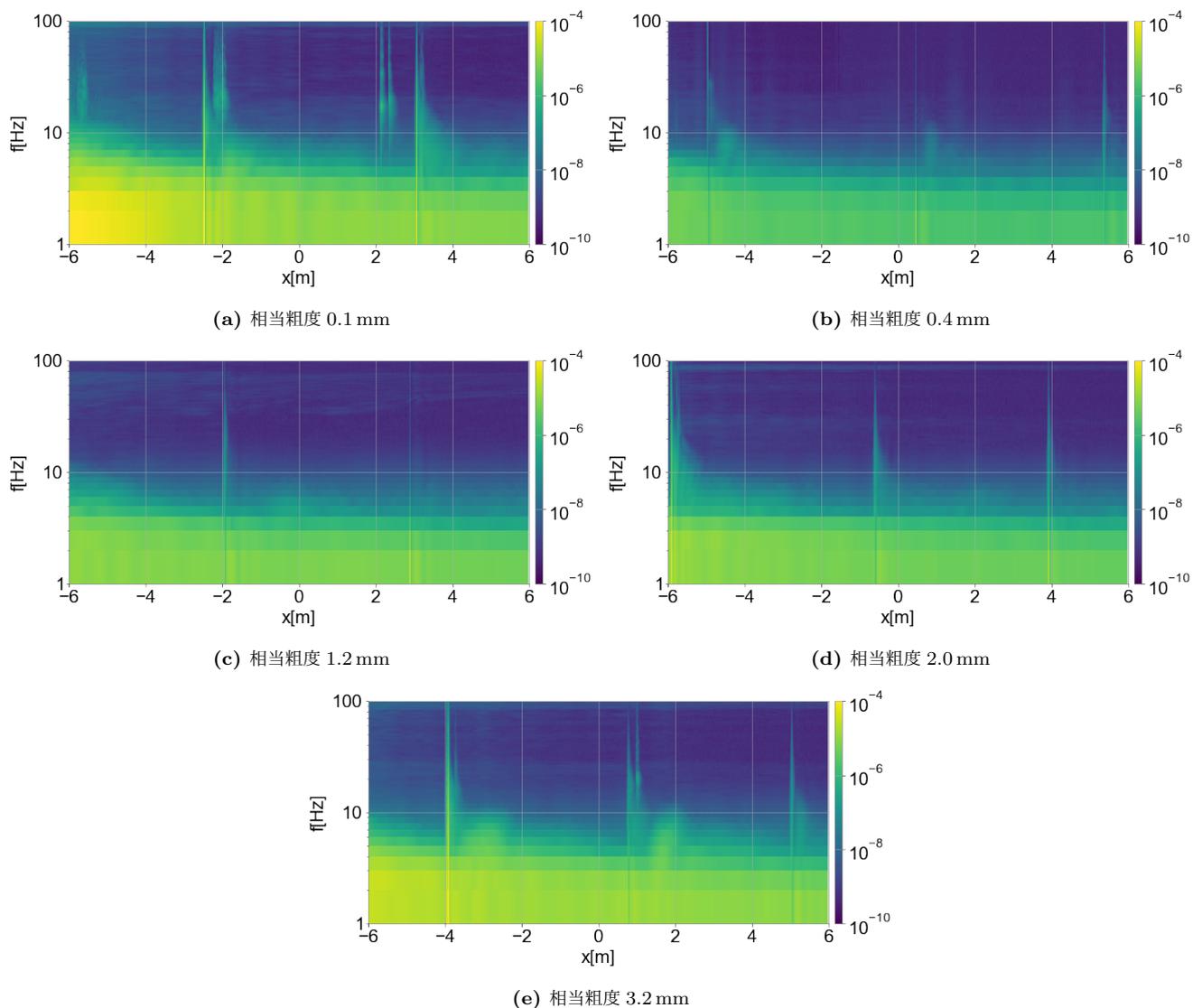


図 5.20: 各条件において、パイプ中心 ( $y = 0$  m) の線上で  $x = -6$  m から  $x = 6$  m の地点までの水面のゆらぎの比較。縦は周波数、横は  $x$  方向の座標、色はゆらぎの強度を表す。1000 秒から 1010 秒の水面のデータを使用している。0.4 mm、1.2 mm、2.0 mm の条件では特異変動水面の 10 Hz 以上のゆらぎが小さいことが分かる。一方 0.1 mm、3.2 mm の条件では 10 Hz 以上においてもゆらぎが大きい特異変動水が 3 箇所以上確認できる。

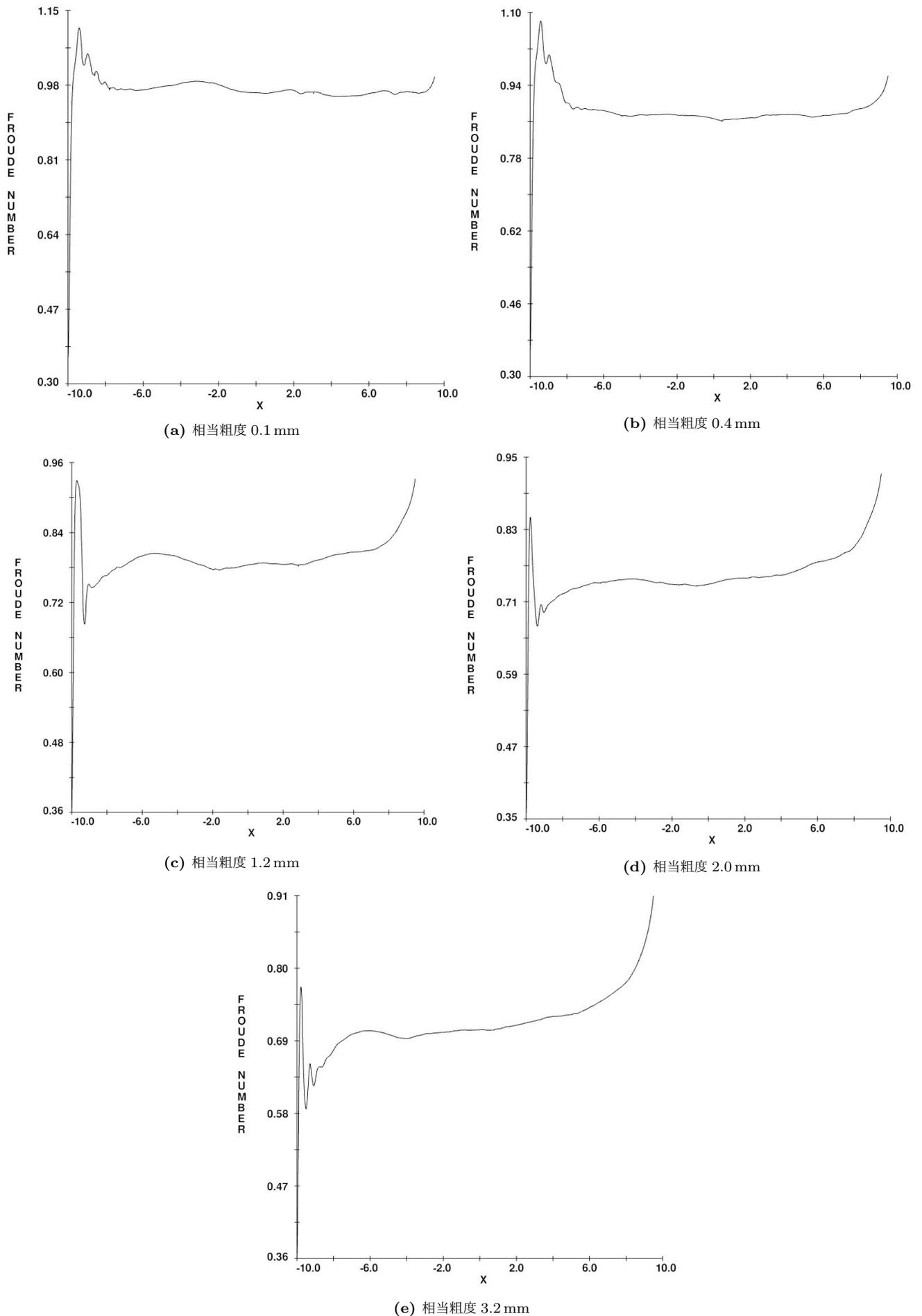


図 5.21: 各条件において、パイプ中心 ( $y = 0$  m) の線上のフルード数の  $x$  方向への分布。相当粗度の値が小さいほど全体的なフルード数は大きくなる。相当粗度 0.1 mm の条件ではフルード数が 1 に近く跳水が発生しやすい状態であると言える。また、他の条件は常流の状態であると言える。

### 5.6.3 傾斜との関係

傾斜の違いの比較においても特異変動水面のゆらぎを調べるために 10 Hz 以上の水面のゆらぎを見た (図 5.22 上部)。これを見ると、どちらの傾斜においても 100 Hz までゆらぎが大きい特異変動水面が数地点で確認できるが、その発生場所が異なっていることが分かる。特異変動水面の数は大きく傾斜 0.3 % の条件で 3 つ、傾斜 0.13 % の条件で 2 つだと確認できる。よって Y-end の傾斜である 0.13 % の条件のほうが特異変動水面の数は少なく、実際の KAGRA の Y-end のパイプにおいても曲がり角の影響を考慮しない場合、Y-end に入る前までのパイプより特異変動水面は少ないと考えられる。よって、ここまでの考察で初期流量や相当粗度、傾斜という 3 つのパラメータを変えた場合に特異変動水面の発生場所やゆらぎの大きさが変化し、明確な相関もないため特異変動水面をあらかじめ狙った場所に出現させることは難しいと考えられる。実際の KAGRA のパイプにおいては初期流量のみある程度コントロールすることができるが、初期流量との相関も見られないため実際の KAGRA のパイプにおいて特異変動水面を回避するように状況をコントロールすることは難しいと考えられる。特異変動水面をコントロールすることができればテストマスをなるべく離し、高周波側の重力勾配雑音に与える影響を小さくすることができるが、現状ではそれは難しいということである。

#### 跳水であるかの考察

次に傾斜の違いによる比較においても Flow-3D から出力されたフルード数の分布 (図 5.22 下部) を確認し、特異変動水面は跳水であるかを確認した。これを見ると、出口に向かってフルード数が大きくなるという傾向は同様であるが、傾斜 0.13 % の方では 0.3 % で見られるようなフルード数の山は見られない。また、このフルード数を信頼する場合どちらの条件においても常流の状態である。よって傾斜の違いで見える特異変動水面は跳水ではないと考えられる。

ここまで直線状パイプの特異変動水面についてフルード数を用いて確認した。一部の特異変動水面においては跳水である可能性を指摘することができるが、フルード数を見た時に跳水であることを否定できる特異変動水面がほとんどであった。よって KAGRA で使用されるパイプにおいては跳水以外の原因でも 10 Hz 以上で大きなゆらぎを持つ特異変動水面が発生可能性があると言える。

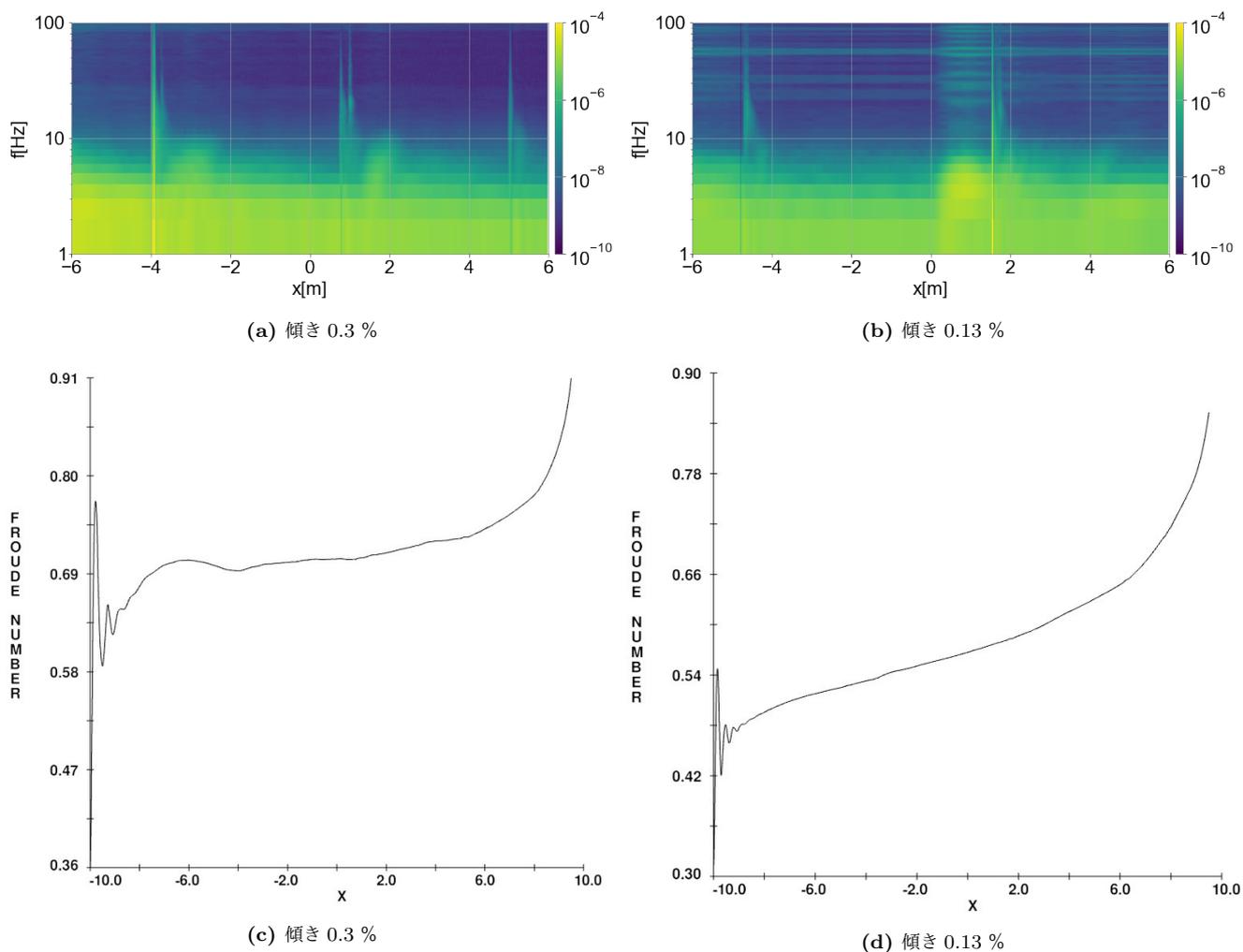


図 5.22: 上部:パイプ中心 ( $y = 0$  m) の線上で  $x = -6$  m から  $x = 6$  m の地点までの水面のゆらぎの比較。縦は周波数、横は  $x$  方向の座標、色はゆらぎの強度を表す。1000 秒から 1010 秒の水面のデータを使用している。どちらの傾斜でも特異変動水面を確認できる。下部:パイプ中心線 ( $y = 0$  m) における  $x$  方向のフルード数の分布。傾斜 0.13 % の条件ではフルード数が単調増加しているため、跳水が起こっていないと言える。また、どちらの傾斜においてもフルード数が 1 以下であるため常流であると言える。

## 第 6 章

# KAGRA のパイプ配置における地下水のシミュレーション

この章では KAGRA のパイプ配置 (Y-end のパイプ) を再現したパイプの地下水のシミュレーションの目的と設定内容について説明し、シミュレーションを解析して得られた水面のゆらぎや水流についての考察を行う。解析の結果得られた重力勾配雑音については 7 章で考察する。

### 6.1 シミュレーションの内容

Y-end におけるパイプの配置を再現したパイプにおいて初期流量を変えて 3 つのシミュレーションを行った (表 6.1)。実際の KAGRA のパイプでは季節変化などにより流量のみが変化する値であるので、地下水の水量の変化と重力勾配雑音の大きさの関係を調べるために初期流量を変えた比較を行った。さらに、相当粗度については実際の値が確定していないが、相当粗度を比較した直線状のパイプのシミュレーションにおいて本論文のシミュレーションで設定した条件では、極端に KAGRA の感度を制限するような重力勾配雑音になる条件がなかったため (重力勾配雑音については 7 章参考) 理論値である相当粗度 3.2 mm を採用した。このシミュレーションは KAGRA のパイプの状況に近いものとなっており、実際の重力勾配雑音と近い解析結果が得られることが期待される。また、このシミュレーションではメッシュサイズ 1.5 cm でシミュレーションを行ったところ膨大な時間<sup>\*1</sup>がかかることが判明したため、メッシュサイズを 2 cm にしてシミュレーションを行った。

KAGRA のパイプを再現したパイプのシミュレーションでは時間的制約により本論文では 3 つの条件 (初期流量 36.2 t/h、109 t/h、181 t/h) の違いのシミュレーションを比較するのみとなった。KAGRA のパイプを再現したパイプのシミュレーションのさらなる比較はこれからの課題としたい。

表 6.1: KAGRA のパイプ配置における地下水のシミュレーションにおける設定パラメータとデータについてまとめ

番号	メッシュサイズ [cm]	傾き [%]	シミュレーション時間 [s]	相当粗度 [mm]	初期流量 [t/h]
K1	2	0.13	1010	3.2	36.2
K2	2	0.13	1010	3.2	109
K3	2	0.13	1010	3.2	181

<sup>\*1</sup> Flow-3D ではその時点の計算速度 (あるタイムステップを計算するのにかかる時間) からシミュレーションの残り時間が予測される。メッシュサイズ 1.5 cm で行ったところシミュレーション終了予測時間は伸びていき、60 日に達していた。計算速度が遅くなればさらに時間がかかりシミュレーションが終了しない恐れもあったため、メッシュサイズ 1.5 cm でのシミュレーションは断念した。

## 6.2 シミュレーションの解析結果に関する議論

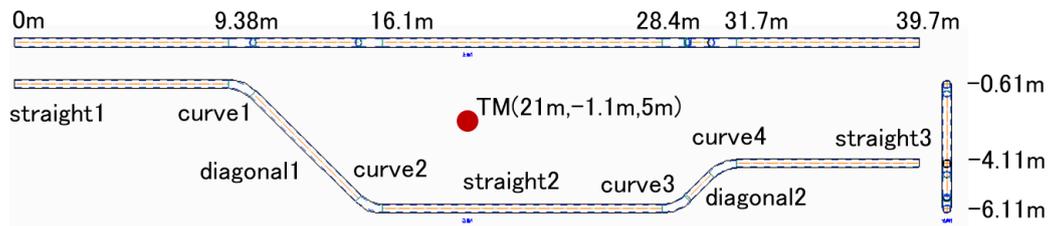


図 6.1: KAGRA の Y-end のパイプにそれぞれの箇所に便宜上名称をつけた。以後この図で示される各場所の名称を用いて記述を行う。

説明の便宜上、図 4.8 を参考に入口から最初の真っすぐな部分を straight1、最初のカーブを curve1、その後の斜めの部分を diagonal1、その後のカーブを curve2、その後の真っすぐな部分を straight2、その後のカーブを curve3、その後の斜めの部分を diagonal2、その後のカーブを curve4、その後の真っすぐな部分を straight3 と呼称する図 6.1。

また、表 6.2 にシミュレーションにかかった時間とデータサイズをまとめた。これを見ると、初期流量が多くなるほどシミュレーション時間は多くなるということが分かる。さらに、データサイズは直線パイプよりも 2 GB ~ 4 GB 大きくなった。186 GB に対してこの差は小さいがその理由としてこのデータファイルはグラフィックを再現するデータ(シミュレーションを視覚的に再現するためのデータ)が多くを占めているという可能性が考えられる。視覚的なデータでは直線パイプと KAGRA のパイプ配置を再現したパイプで差が少ないため、結果的に直線パイプとほぼ同じデータサイズになっていると考察できる(つまり、グラフィックを除いたシミュレーションのデータは数 GB の可能性がある)。なお、解析に必要なテキストデータのサイズは大きいので、データを分割するなどの工夫が必要であった。

表 6.2: Y-end のパイプを再現したパイプにおけるシミュレーションのシミュレーションにかかった時間やシミュレーションのデータサイズ

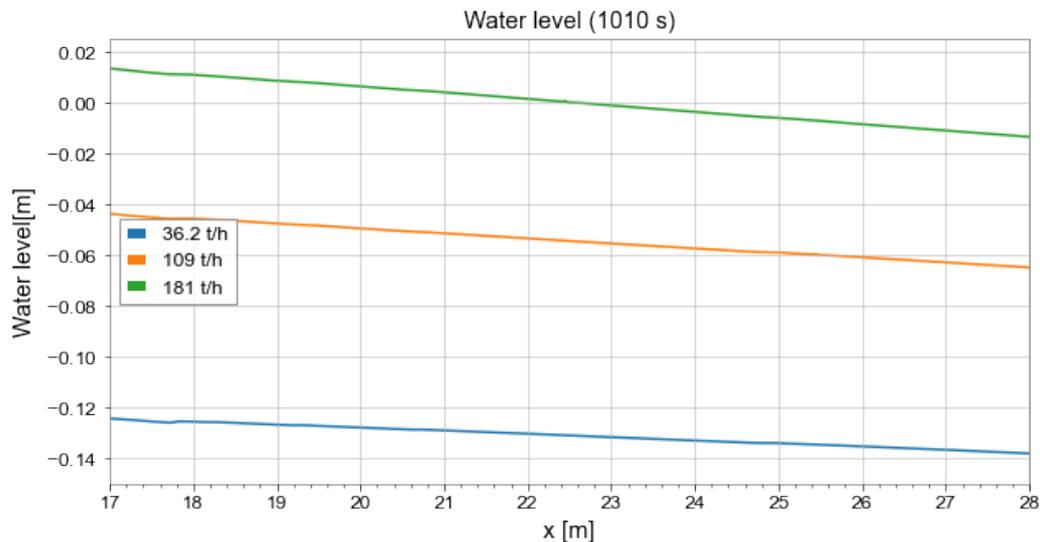
番号	初期流量 [t/h]	シミュレーション時間	シミュレーションにかかった時間 [s]	データサイズ [GB]
K1	36.2	1010 秒	388783	186
K2	109	1010 秒	456300	186
K3	181	1010 秒	467260	186

### 6.2.1 水面のゆらぎ

Y-end のパイプを再現したパイプのシミュレーションにおいても、水面の様子を観察した。ただし、曲がり角などは全体のゆらぎの把握が難しく、重力勾配雑音への影響が大きいテストマスに近い直線部分 (straight2) のゆらぎを中心に解析した。直線状パイプのシミュレーションにおいて水面には局所的に大きなゆらぎを持つ特異変動水面が現れることが判明した。そのため、KAGRA のパイプ配置における地下水のシミュレーションにおいても特異変動水面が現れるかを確認する。なお、傾斜の違いの比較では傾斜 0.13 % の条件のほうが特異変動水面の発生個数が少なく直線状パイプよりも特異変動水面は発生しにくいと考えられる。ただし、Y-end のパイプには曲がり角があり、その影響により特異変動水面が現れる可能性がある。

図 6.2 は各条件における straight2 のパイプ中心線  $y = -6.1 \text{ m}$  に沿った  $x$  方向の水位である。これを見る直線状パ

イプと同様に上流のほうが水位が高く、下流に向かうにつれて水位が減少していくことが分かる。また、直線状パイプの初期流量比較で見られた傾向と同様に初期流量が少ないほど水位も低くなること分かる。図 6.3 は straight2 にお



(a) 初期流量 36.2 t/h

図 6.2: 各条件において、straight2 の  $x$  方向へのパイプの中心線に沿った等間隔の地点における水位。シミュレーション終了時間の 1010 秒時。どの条件においても下流方向に水位が低くなっていくことが分かる。また初期流量が少ないほど、水面の高さが低いことも分かる。

ける  $x$  方向へのパイプ中心線に沿った等間隔の地点の水面のゆらぎである。これを見ると 1 Hz~2 Hz でゆらぎのピークがあることが分かる。このピークは straight2 の全体でみられる。初期流量 36.2 t/h の条件ではこのピークの高さは低周波側のゆらぎの大きさより小さいが初期流量 109 t/h、181 t/h の条件では 10 Hz までのゆらぎの中で最も大きいゆらぎとなっている。よって straight2 はテストマスに最も近い領域であるため、1 Hz~2 Hz のゆらぎのピークは重力勾配雑音にも大きく影響を与え、109 t/h、181 t/h ではピークとして現れると予想される。さらに、場所によっては 2 Hz~10 Hz においても鋭くはないがこの帯域全体にかかるようなピークが見られる。よって重力勾配雑音においても 1 Hz~2 Hz に現れるピークよりは小さいと考えられるが、ピークが見える可能性がある。また、このピークは局所的なものではなく、図 6.3 で設定した多くの地点で見られるため、直線状パイプで見たような局所的な特異変動水面とは性質の違うゆらぎであると考えられる。

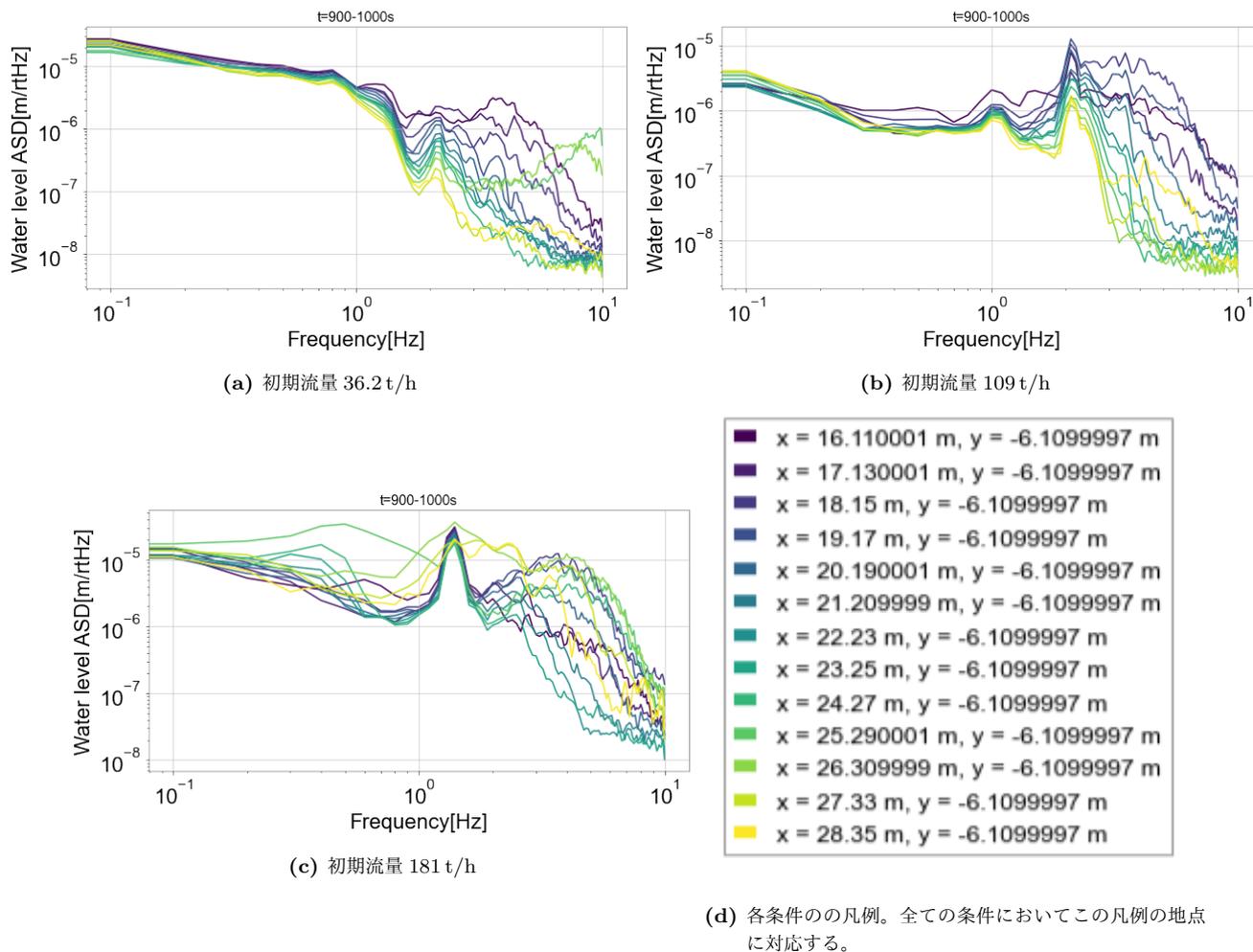


図 6.3: 各条件において、straight2 の x 方向へのパイプの中心線に沿った等間隔の地点における水位のスペクトル。地点は凡例の通りで、色が紫に近い方が上流に近く、黄色に近いものが下流に近い。全ての条件で 1 Hz~2 Hz ピークが見られる。

### 6.2.2 大きなゆらぎを持つ地点の考察

さらにピークとなる水面のゆらぎを詳しく見るために、straight2 のパイプ中心線に沿う全地点のゆらぎ図 6.4 を見る。109 t/h、181 t/h の条件では大部分の地点で 1 Hz～2 Hz でゆらぎの大きいピークを持つことが分かる。よって、特定の地点で見えている現象ではないので直線状パイプのシミュレーションで見られた特異変動水面ではなく、また跳水ではないと考えられる。このピークは straight2 全体に出ているためパイプの構造に由来するものである可能性が高い。パイプの構造に着目した時に考えられる可能性として 2 つ考えられる。まず、考えられる可能性はパイプの壁面か

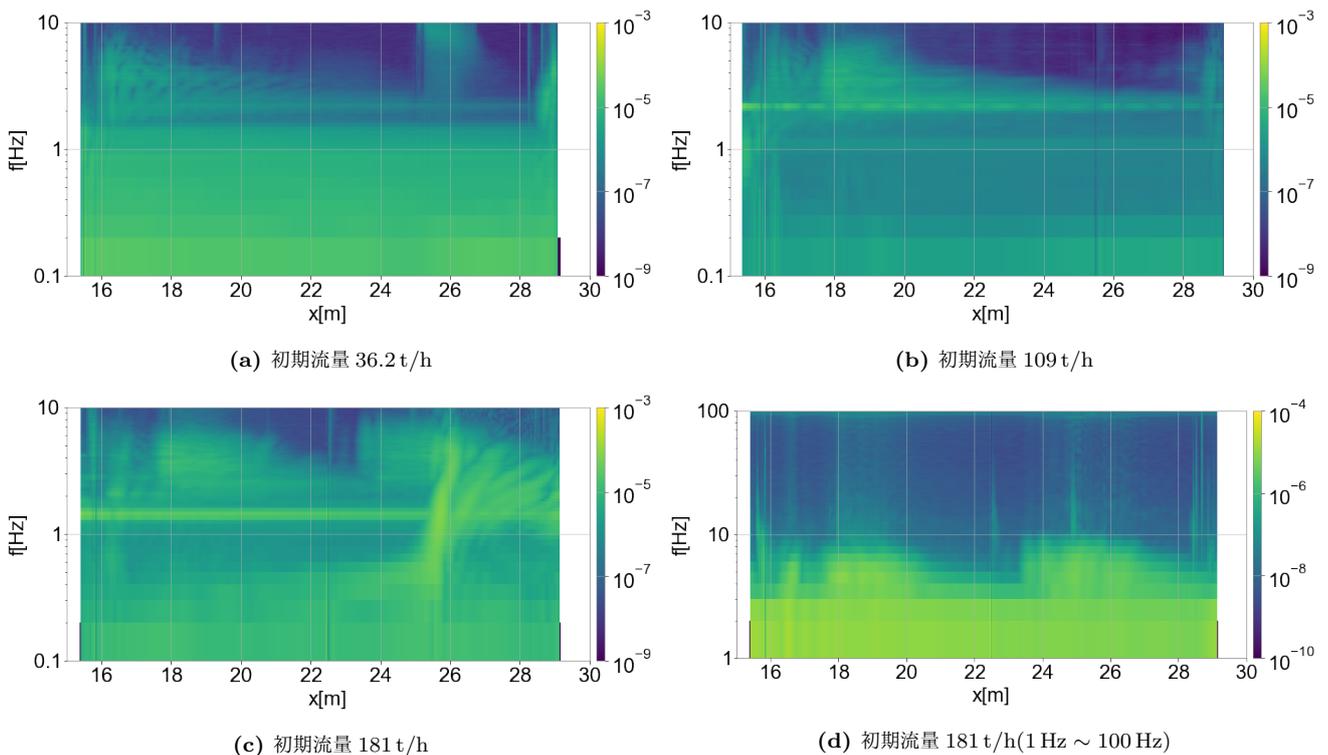


図 6.4: 各条件において、straight2 の  $x$  方向へのパイプの中心線に沿った地点の水面のゆらぎの比較。縦は周波数、横は  $x$  方向の座標、色はゆらぎの強度を表す。

らの影響である可能性である。straight2 におけるパイプ壁面で 1 Hz～2 Hz のゆらぎが発生した場合全域でこのピークが見える。

次に考えられる可能性は直線状パイプのシミュレーションではこのピークが見られなかったため straight2 前後にある curve3、curve4 が原因である可能性である。直線状パイプのシミュレーションにおいては入口と出口の影響がそれぞれ 4 m 前後の領域であり、(入口と出口から 4 m を除いた水面を見ていたが、この範囲では入口と出口からの影響は限定的であった)、straight2 の長さは約 12 m であることを考えると、straight2 のカーブに近い場所では曲がり角の影響があると考えられる。また、181 t/h の条件で 26 m 付近で局所的な大きなゆらぎが発生していることが分かる。この揺らぎは直線状パイプの特異変動水面とは異なり、0.3 Hz～8 Hz 付近でゆらぎが大きい。このゆらぎは跳水である可能性と、曲がり角の直前で発生しているため曲がり角の影響の可能性が考えられる。

### パイプ壁面からの影響

パイプ壁面からの影響を見るために straight2 の  $y$  方向への水面のゆらぎを見た (図 6.5)。これを見ると 1 Hz~2 Hz のピークは壁面で特に大きくなく  $y$  方向全てで見えていることが分かる。よってこのピークはパイプ壁面に由来するものではない。

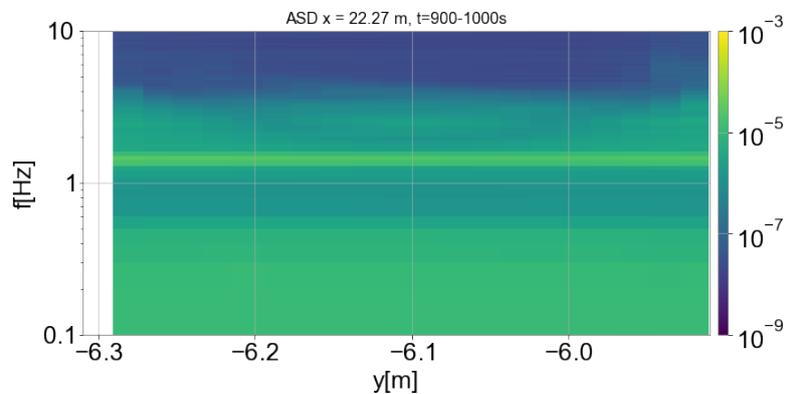


図 6.5: 各条件において、straight2 の 22 m 地点の  $y$  方向への水面のゆらぎの比較。縦は周波数、横は  $y$  方向の座標、色はゆらぎの強度を表す。1 Hz~2 Hz のゆらぎはパイプ壁面で大きいわけではなく横方向全体にも出ていることが分かる。

### 曲がり角のゆらぎ

直線部分のように全体のゆらぎの把握は難しいが初期流量 181 t/h の条件で curve1、curve2、curve3、curve4 のうち 1 地点のゆらぎを観察した (図 6.6)。これを見るとどのカーブにおいても 1 Hz~4 Hz 付近でゆらぎのピークがあることが分かる。よって straight2 部分だけでなく、カーブ部分の水のゆらぎも重力勾配雑音においてこの帯域のピークを作る原因となる可能性がある。また、curve2 において 1 Hz~2 Hz のピークが発生しているため、curve2 で発生した 1 Hz~2 Hz のゆらぎが straight2 へ伝搬している可能性が考えられる。そこで初期流量 181 t/h の条件で curve2 の地点 (straight2 の入口) と straight2 のコヒーレンスを調べた (図 6.7)。これを見ると 1 Hz~2 Hz で straight2 の 25 m 付近までコヒーレンスが高いことが分かり、curve2 で発生した 1 Hz~2 Hz のゆらぎが伝搬している可能性が高いことが考えられる。

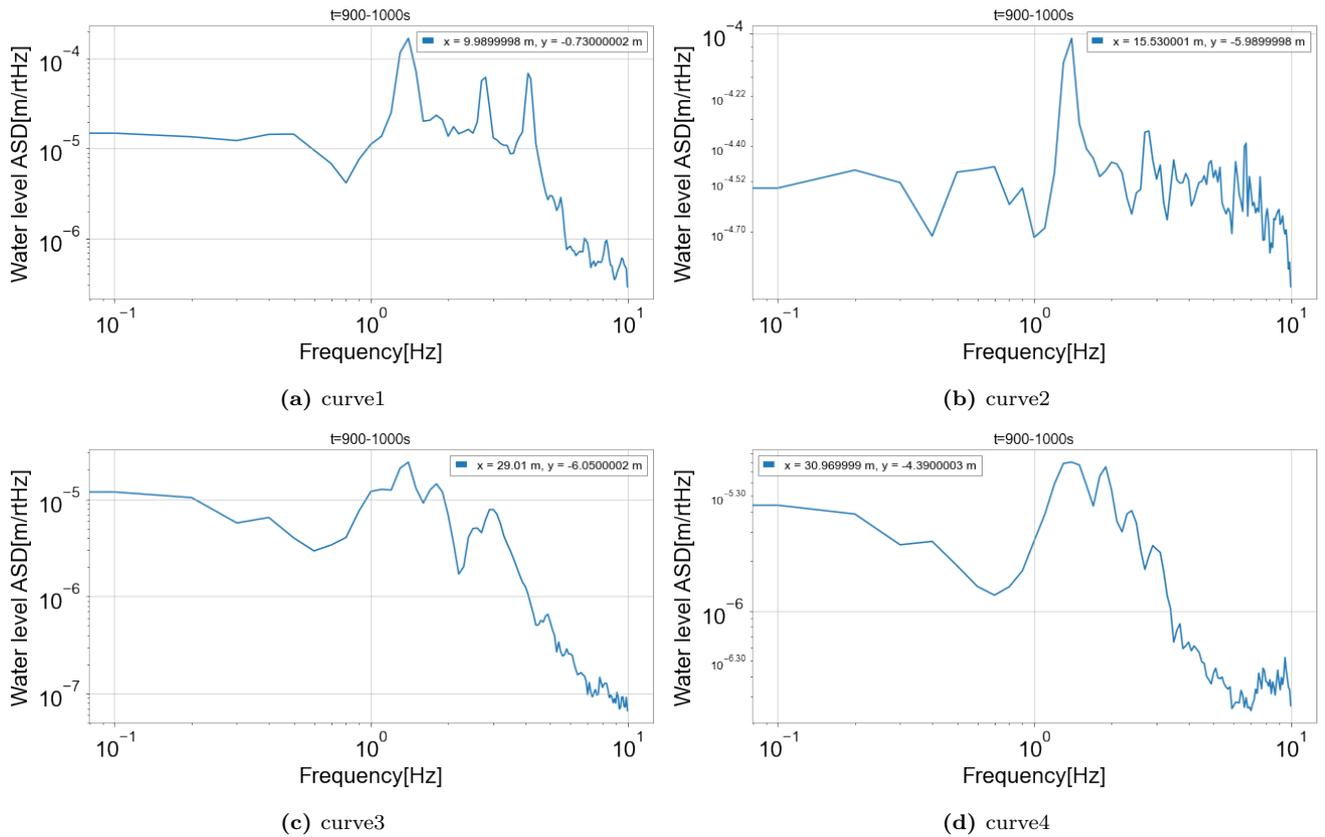


図 6.6: 初期流量 181 t/h の条件で各カーブ内の 1 地点のゆらぎ。全てのカーブで 1 Hz~2 Hz のゆらぎが大きいことが分かる。

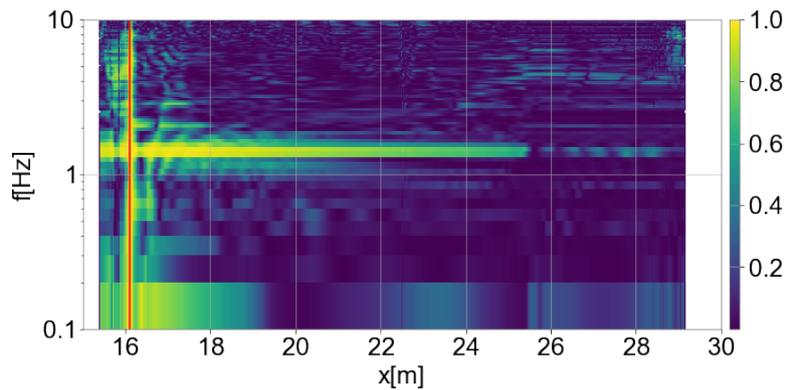


図 6.7: 初期流量 181 t/h の条件で curve2 の終了地点 16 m と straight2 の ( $y = -6.1$  m) の全域のコヒーレンス。1 Hz~2 Hz で 25 m 地点までコヒーレンスが高いことが分かる。

### 跳水であるかの考察

straight2 で直線状パイプで見たような特異変動水面は確認できなかったが 181 t/h の条件で 10 Hz 以下のゆらぎが大きい特異変動水面が見られたため、跳水であるかを調べる。跳水が起きている可能性があるか確かめるためにフルード数を調べた (図 6.8)。その結果、全ての条件でフルード数は多くの部分で約 0.4 であり、常流の状態であることが分かった。さらに、跳水が起こる際に見えるフルード数の低下も、特異変動水面がある 181 t/h の 26 m 付近では見られていない。よってこの地点の特異変動水面は、跳水ではない可能性が高い。なお、181 t/h の条件で 26 m 付近で見えていたゆらぎは跳水ではないがパイプが曲がっている構造に由来している可能性が高い。

### 6.2.3 パイプの断面・ $x$ 方向の流速

KAGRA のパイプ配置における地下水のシミュレーションにおいて straight2 部分の流速の分布を調べた (図 6.9)。straight2 入口付近では一時的に流速が上昇していることが分かる。またどの条件においても、その後下流に向けて、流速が速くなっている。また、断面の分布をみると初期流量が多いほど水量が多いことも分かる。断面の分布では他のシミュレーションと同じ様に壁面の流速が遅い。さらに断面の速度分布を見ると、 $y$  方向負の側に流速が速い部分が偏っていることが分かる。これは直線状パイプのシミュレーションでは見られなかった傾向であり直線状パイプにはなかった曲がり角の影響だと考えられる。このように速度が偏っているのは diagonal1 が  $y$  方向負の側に向いており、curve2 を通って straight2 に流入するためだと考えられる。つまり、curve3 断面を見た時にカーブの外側は  $y$  方向負の側であり curve3 までの水が straight2 の  $y$  方向負の側に流れやすいということである。また、断面が滑らかではないが、これは Flow-3D の出力の仕様である。しかし、ここまでの粗さはないものの、パイプの円は CAD ソフトの性質上完全に円ではないことに注意したい。

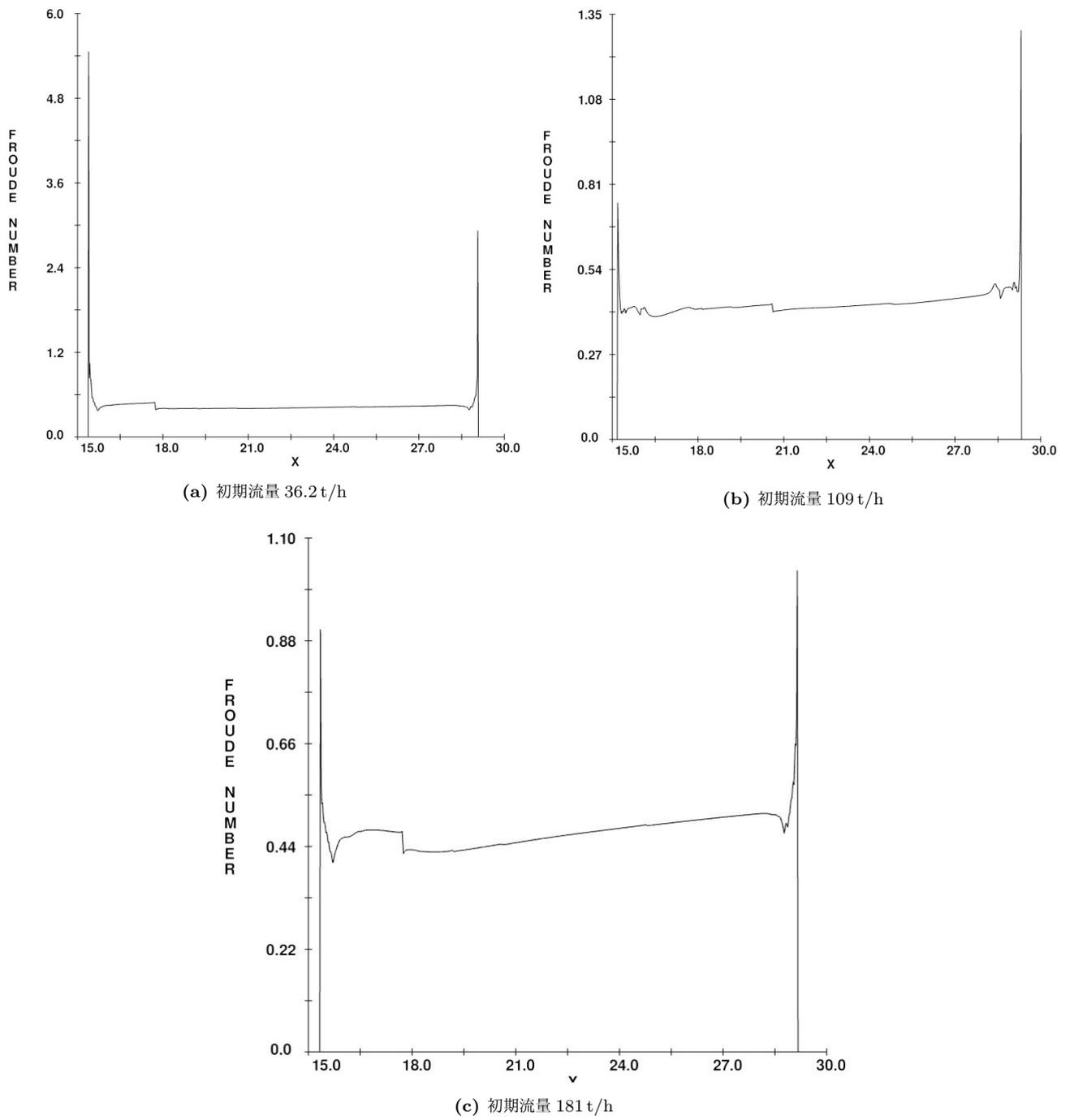
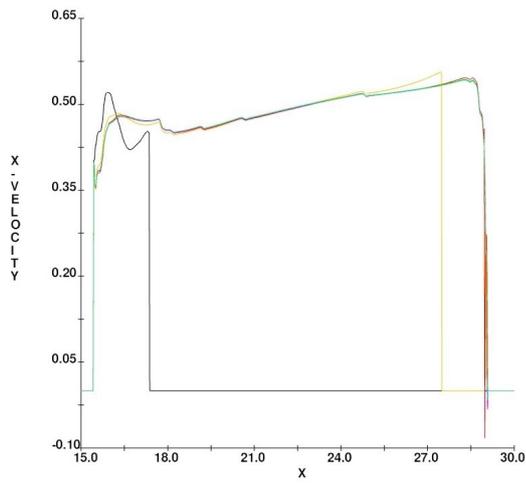
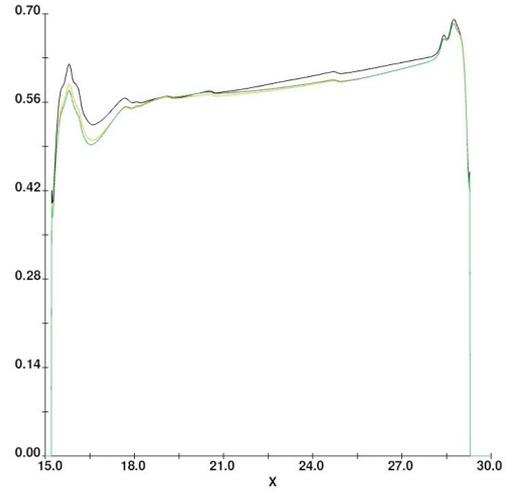


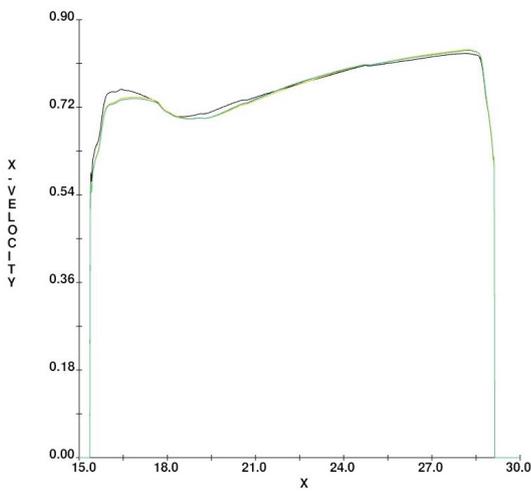
図 6.8: 各条件において、straight2 のパイプ中心線上のフルード数の x 方向への分布。



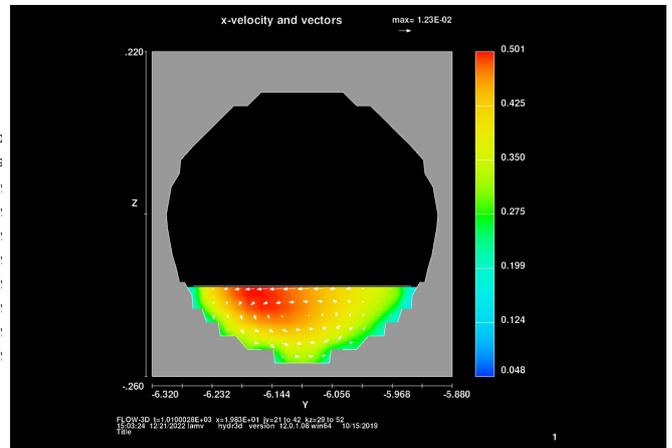
(a) 初期流量 36.2 t/h



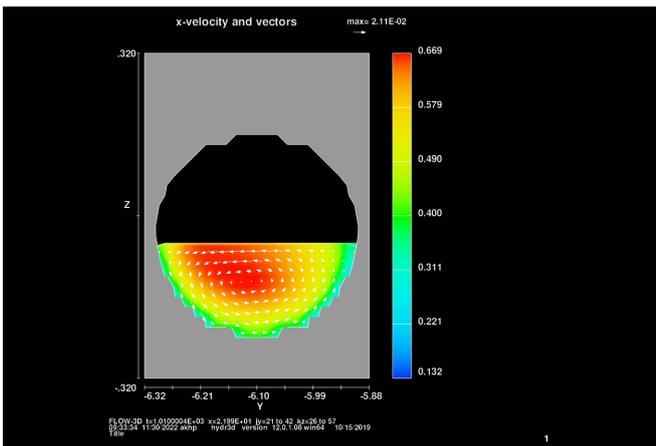
(b) 初期流量 109 t/h



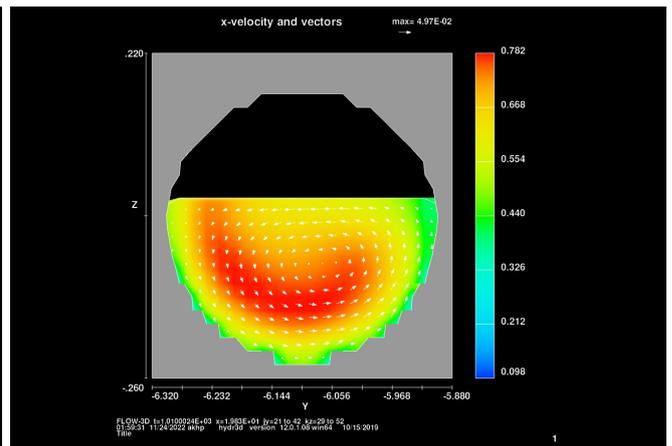
(c) 初期流量 181 t/h



(d) 初期流量 36.2 t/h



(e) 初期流量 109 t/h



(f) 初期流量 181 t/h

図 6.9: 各条件において straight2 のパイプの中心線で、水面に近い位置の  $x$  方向の流速の大きさの  $x$  方向の分布と straight2 の断面の流速の分布。流速に偏りがあることが分かる。

## 第7章

# 重力勾配雑音の解析結果

Flow-3Dで行ったシミュレーションのデータを元に得られた重力勾配雑音の結果を説明する。まず、シミュレーションの解析方法(重力勾配雑音の計算)について説明する。本実験では流体シミュレーションの結果をFlow-3Dの機能によりテキストデータとして出力し、プログラミング言語Pythonを使用し解析した。本論文では重力勾配雑音の結果とともにKAGRAの設計感度と第三世代重力波望遠鏡Einstein Telescope(ET)の設計感度([6]内より引用)を載せている。パイプはKAGRA内にあるので、ETの感度について直接に関係があるものではないが、建設される場合にETの鏡とパイプがKAGRAと同関係にあった場合という仮定の比較となる。ETの設計感度を載せた理由としては、ETは地下に建設される予定であり、パイプを通して地下水を排出する必要がある可能性があるからである。また、ETは低周波帯の設計感度がKAGRAよりも良い予定であり、より重力勾配雑音の影響を受けやすいことが予想される。

### 7.1 Pythonを使用した処理

本論文ではFlow-3Dのデータの解析、及び重力勾配雑音の計算をPythonで行った。本研究における水の重力勾配雑音の定式は1章で説明した通りであり、Flow-3Dから出力されたflow depthとterrain elevationのテキストデータをPythonで読み込み重力勾配雑音の計算を行った。Flow-3Dの出力をプログラム上で解析できるようにデータを整え、その後解析するコードを作成した。付録Cに実際に作成、使用したコードを載せる。コードはJupyter notebook上で動く.ipynb形式で書かれている。コードは解析に必要な関数を独自に作成し(C.1前半)、それらを組み合わせ、重力勾配雑音や水面のゆらぎについての解析を行った。

簡単に解析の流れを説明する。まず、各 $x, y$ 座標ごとのflow depthの値とメッシュによって設定された面積(低面積)をかけ、水の体積を算出した。その後、flow depthの半値とterrain elevationの値を足し、その値(HalfLevelData)をその座標における、水の重心座標とした。なお、それぞれ各時系列ごとに計算を行い、最終結果は時系列データとして算出される。こうして算出した水の体積(質量)と重心座標を式(1.32)に代入し、鏡にかかる加速度の時系列データを算出した。重力勾配の計算で行う、万有引力を足し合わせる積分にはPythonの台形積分(integrate.trapz)を用いた。図7.1はこの計算のイメージ図である。その後Pythonで使用できるパッケージGWpy[23]の信号処理のメソッドを使い周波数領域における特性を調べた。

#### 7.1.1 重力勾配雑音の計算

重力勾配雑音の計算部分のコードを解説する。

Code 7.1: 重力勾配雑音の計算部分の抜粋

```
1 #ニュートニアン計算
```

```

2     def Newtonian(hyfd, HalfLevelData, ix_min=0, ix_max=0, iy_min=0, iy_max=0, t_start=900,
3         t_end=1000):
4         G=6.67408*10**(-11) #万有引力定数
5         rho=1000
6         x_TM=0
7         y_TM=5
8         z_TM=1.5 #[m]
9         R=0.2 #[m]
10        data=copy.deepcopy(hyfd.data)
11        times=copy.deepcopy(hyfd.times)
12        MBs=copy.deepcopy(hyfd.MBs)
13        unit=copy.deepcopy(hyfd.unit)
14        data2=copy.deepcopy(HalfLevelData.data)
15        times2=copy.deepcopy(HalfLevelData.times)
16        MBs2=copy.deepcopy(HalfLevelData.MBs)
17        unit2=copy.deepcopy(HalfLevelData.unit)
18        ACC=np.zeros(times.size)
19        i_MB=0
20        #     for i_MB in range(len(MBs)):
21        if ix_max==0:
22            ix_max=MBs[i_MB].nx-1
23        if iy_max==0:
24            iy_max=MBs[i_MB].ny-1
25        for i_t in range(times.size):
26            r3=((x_TM-MBs[i_MB].xx)**2+(y_TM-MBs[i_MB].yy)**2+(z_TM-data2[i_t][i_MB])
27                **2)**1.5
28            dACC=G*rho*data[i_t][i_MB]*(MBs[i_MB].xx-x_TM)/r3
29            ACC[i_t] = integrate.trapz(integrate.trapz(dACC[iy_min:iy_max+1,ix_min:
30                ix_max+1], MBs[i_MB].y[iy_min:iy_max+1], axis=0), MBs[i_MB].x[ix_min:
31                ix_max+1])
32            ACC_time=TimeSeries(ACC, t0=hyfd.t0, dt=hyfd.dt, unit='m/s^2')
33            ACC_ASD=ACC_time.crop(t_start, t_end).asd(1, 0.5)
34            Strain_ASD=ACC_ASD/(2*pi*ACC_ASD.frequencies)**2/3000
35        return ACC_time, ACC_ASD, Strain_ASD

```

Code7.1は重力勾配雑音を計算するために作成した関数である。入力には水深(hyfd)、各地点の水の重心座標(HalfLevelData)と積分範囲(ix\_min、ix\_max、iy\_min、iy\_max)、使う時間(t\_start、t\_end)である。積分範囲は入力しなければ、入力された水深の全範囲を積分範囲とする。この関数ではまず、計算に必要な万有引力定数(G)、水の密度(rho)、テストマスの座標(x\_TM、y\_TM、z\_TM)、パイプ半径(R)を設定している。その後の25行目からが重力勾配雑音の計算となっている。25行目のfor文は(times.size)で計算を繰り返しているが、これは重力勾配雑音の計算を各時間ごとに行っていることを示す。つまり、各時間の水からの重力相互作用によって受けるテストマスの加速度を計算している。この計算は式(1.32)を計算し、 $a$ の値を出すものである。このテストマスの加速度の計算は、まず26行目でテストマスと重心座標から式(1.32)の被積分関数の分母( $r_3$ )を計算している。MBs[i\_MB].xx、MBs[i\_MB].yy、data2[i\_t][i\_MB]は水の重心座標を表し、全ての地点の水の重心座標が格納されている。そして27行目では被積分関数(dACC)を計算している。28行目では計算された被積分関数(dACC)を対象にPythonにおける台形積分 integrate.trapz を使用し、指定された範囲で  $x$ 、 $y$  方向への積分を行っている。このようにある時間のテストマスの加速度(ACC[i\_t])が計算され、この計算が時間によるfor文で繰り返されることにより、テストマスが受ける加速度の時系列データ(ACC)が計

算される。その後 ACC を Python の信号処理パッケージ GWpy[23] で解析している。具体的には 29 行目で GWpy における時系列データにし、30 行目で GWpy のメソッドである `.asd` により、時系列データを振幅スペクトル密度に変換している。`.asd` は `.asd(fftlength, overlap, window='hann')` であり、この例では `fftlength=1` 秒、`overlap=0.5` 秒、窓関数=hann 窓 (デフォルト) となっている。これはつまり 1 秒の時間幅で 0.5 秒ずつずらしながら、窓関数は hann 窓を使い高速フーリエ変換を行い、それを平均化して計算結果 (振幅スペクトル密度) を出していることを示す。31 行目では重力波信号の次元に重力勾配雑音を変換している。32 行目ではこの関数の出力を示し、重力勾配雑音の時系列データ (ACC\_time)、重力勾配雑音の振幅スペクトル密度 (ACC\_ASD)、重力勾配雑音の重力波信号の次元のもの (Strain\_ASD) が出力される。

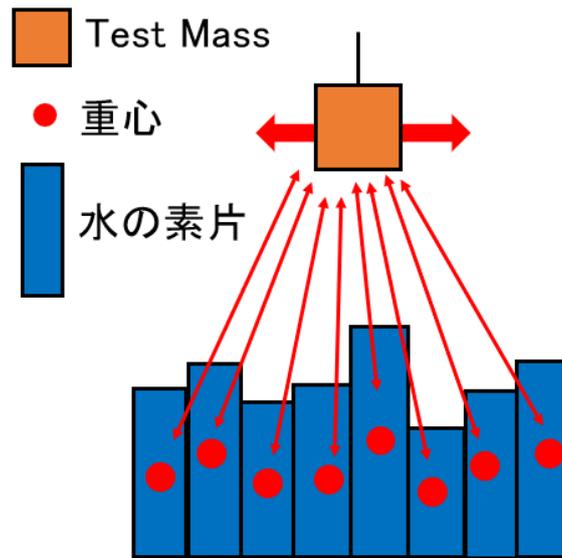


図 7.1: 水を素片に分け、テストマスがそれぞれの素片から受ける力を足し合わせ加速度を計算した。パイプ表面の高さ座標に水深の半分の長さを足すと水の重心位置となる。

### テストマスの位置の設定

Code7.1 内でテストマスの座標を入力している。 `x_TM`、 `y_TM`、 `z_TM` と入力している。この例では (0 m、5 m、1.5 m) となっているが、これは直線状パイプにおける設定である。真っすぐなパイプではパイプの進行方向を  $x$  軸正方向として、重力と  $x$  軸に垂直な方向を  $y$  軸方向、重力に対して負の方向を  $z$  軸正方向としている。パイプの中心が原点である (パイプの入口から +10 m、出口から -10 m の位置で、パイプの断面の円の中心)。この座標系で、テストマスの座標を決定している。このテストマスの位置は実際の KAGRA のパイプの位置関係図 2.3 を参考にしている。

一方 Y-end のパイプを再現したパイプでは図 2.3、図 2.4 の位置関係を参考にしている。具体的な値としては、 `x_TM=21` m、 `y_TM=-1.1` m、 `z_TM=1.5` m である。パイプ入り口から最初の曲がり角までの部分を基準に、直線状パイプと同様の  $x$ 、  $y$ 、  $z$  方向となっており、原点はパイプ入り口の付近である (パイプ入り口部分の断面の中心の座標は (0.01 m、0 m、-0.6 m))。

## 7.2 直線状パイプにおけるシミュレーション

直線状パイプのシミュレーションの結果を説明する。4章で説明したように、10 Hz 以下の重力勾配雑音の算出にはシミュレーションの900秒～1000秒の水面のデータを使用した。100秒の重力勾配雑音の時系列データを出した後に、10秒の時間幅、5秒間隔でPython上で高速フーリエ変換を行い平均化した。10 Hz～100 Hzの重力勾配雑音は1000秒～1010秒の水面のデータを使用し、10秒の重力勾配雑音の時系列データを出した後に、1秒の時間幅0.5秒間隔で高速フーリエ変換を行った。重力勾配雑音の結果はこれら2つの時間から計算されたものを繋げて表示している。また重力勾配雑音の算出には、 $x$ 方向にはパイプの出口と入口部部分からそれぞれ4mを除いた12m、 $y$ 方向には全領域の水の質量を使用した。

### 7.2.1 初期流量の違いによるシミュレーションの比較

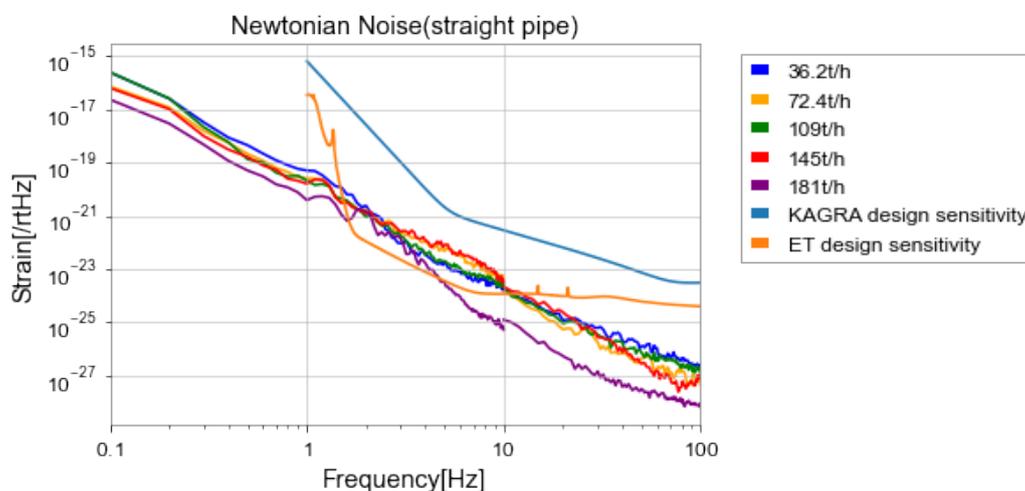
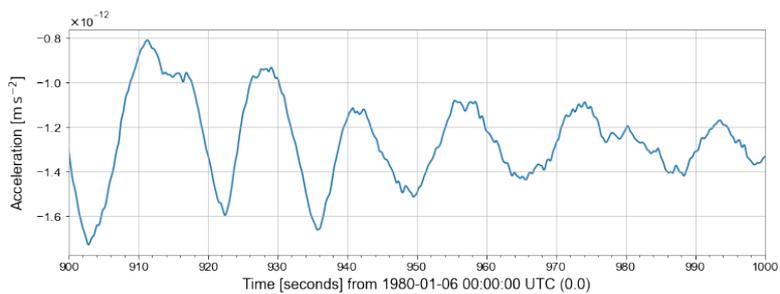
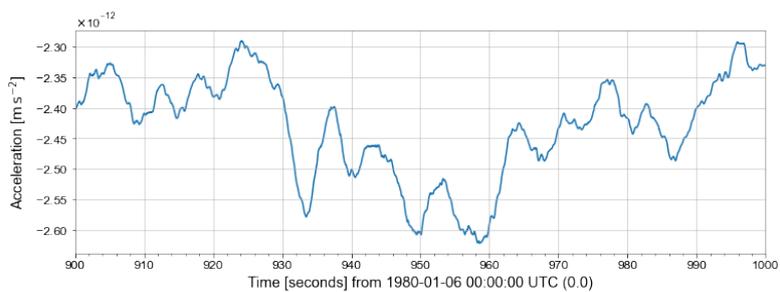


図 7.2: 直線状パイプのシミュレーションの初期流量を変えた場合の重力勾配雑音の比較。比較したシミュレーションは表 5.1 の番号 S1、S7、S13～S15。10 Hz 以下の雑音は 900 秒～1000 秒のサンプリング周波数 20 Hz のデータから計算されたものである。10 Hz～100 Hz の雑音は 1000 秒～1010 秒のサンプリング周波数 200 Hz のデータから計算されたものである。181 t/h の条件で特に重力勾配雑音の大きさが小さいことが分かる。

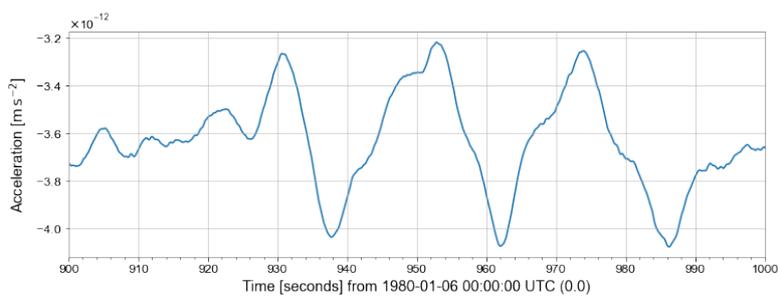
初期流量の違いによるシミュレーションにおける重力勾配雑音についての解析の結果を示す (図 7.2)。また、参考に重力勾配の影響による鏡にかかる加速度の時系列データに関しても載せる (図 7.3)。図 7.2 を見ると、本シミュレーションの条件ではどの初期流量においても水からの重力勾配雑音は KAGRA の設計感度を越えない (1 Hz～100 Hz)。しかし、ET の設計感度を 1 Hz～10 Hz の領域で越えてしまう。さらに各条件を比較すると、初期流量 181 t/h の条件で他の条件に比べて雑音は小さい。特に 4 Hz 付近より高周波側では他の条件との差が顕著である。また、本シミュレーションの条件では初期流量と雑音の大きさに明確な比例関係はないことが分かる。10 Hz 以下では 72.4 t/h と 145 t/h の条件で雑音が大きいが、10 Hz 以上では逆に 36.2 t/h と 109 t/h の条件より下がっている。2 Hz までではどの条件も同程度の雑音である。また、初期流量は現実の KAGRA のパイプでも季節によって大きく変化するものであり、水面を直接変化させる要因であると予想されていた。図 7.2 を見ると初期流量の変化による重力勾配雑音の大きさは KAGRA の感度と比較した場合許容できる範囲の変化と言える。



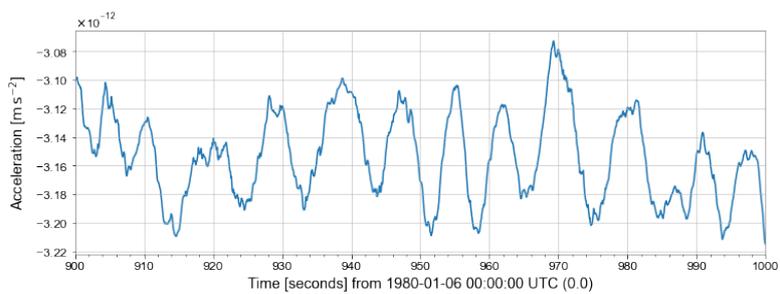
(a) 初期流量 36.2 t/h



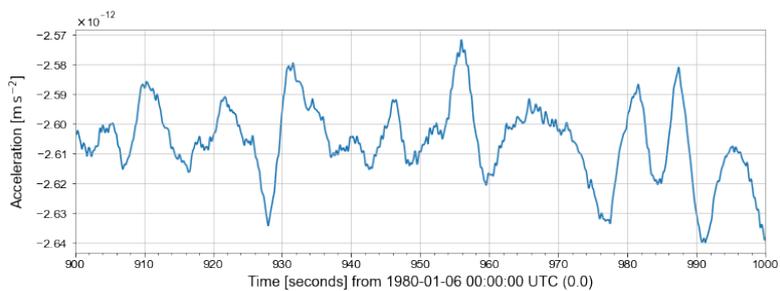
(b) 初期流量 72.4 t/h



(c) 初期流量 109 t/h



(d) 初期流量 145 t/h



(e) 初期流量 181 t/h

図 7.3: テストマスにかかる加速度 (重力勾配雑音) の 900 秒~1000 秒の時間変化。

### 重力勾配雑音と水面のゆらぎとの関係についての考察

次に重力勾配雑音と水面のゆらぎの関係について考察する。まず、重力勾配雑音は低周波側の方が大きく、これは水面のゆらぎ図 5.7、図 5.18 のゆらぎの大きさの傾向とも一致する。水面のゆらぎが重力勾配雑音と直接的な関係にあることが、式 (1.32) のみではなくこの結果からも分かる。

次に初期流量 181 t/h の条件では水面のゆらぎが他の条件よりも小さかったが (図 5.6) 重力勾配雑音の結果 (図 7.2) を見ても、他の条件よりも雑音が小さいことが分かる。これは水面のゆらぎの小ささが重力勾配雑音の比較にも現れていると考えることができる。

次に重力勾配雑音の結果と特異変動水面の関係について考察する。図 7.4 は特異変動水面ではない地点を各条件で比べたものである (それぞれの条件で同座標地点)。特異変動水面ではなく水面の大部分のゆらぎ (から一部分をとった地点) の比較である。これを見ると 20 Hz~100 Hz の周波数帯で 36.2 t/h と 72.4 t/h の条件で大きく、他の条件は同程度に小さいことが分かる。全ての地点でこのような大小関係であるわけではないが、図 5.18 を見るとテストマスに近い大部分の地点で同様の比較が行えると考えられる。また、位相的にもこの地点と他の多くの地点とのコヒーレンスも高い (10 Hz 以下) ため、水面のゆらぎの比較として最適な地点である (図 7.5)。ここで重力勾配雑音の結果図 7.2 に注目する。重力勾配雑音の 10 Hz 以上の部分の比較では特に 181 t/h の条件が小さかった。しかし図 7.4 でその周波数帯では 109 t/h、145 t/h、181 t/h の条件ではほぼ同程度の水面のゆらぎの大きさである。重力勾配雑音が水面のゆらぎそのものから影響を受けている性質上、重力勾配雑音の大きさの差は水面から来ているはずである。よって重力勾配雑音の 10 Hz 以上の部分で見られた差は特異変動水面ではない地点での比較では説明できない。つまり重力勾配雑音の 10 Hz 以上の部分で見られた差は、特異変動水面のゆらぎの大きさの差やそのゆらぎが伝搬している領域の大きさの差、テストマスとの距離の差によって生まれている可能性が高い。他の可能性として、重力勾配雑音は水面のゆらぎの足し算な

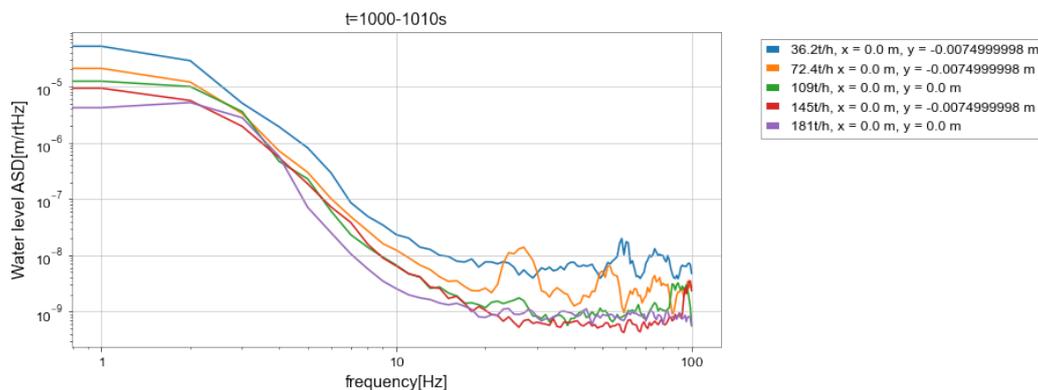


図 7.4: 各条件における  $(x, y) = (0, 0)$  地点の水面のゆらぎの比較。特異変動水面ではない地点の比較として、この地点を選んだ。また、この地点は  $y = 0$  m の線上ではテストマスに最も近い位置であり、重力勾配雑音に関する影響を見る地点として最適である。1000 秒から 1010 秒の水面を解析した結果である。109 t/h、145 t/h、181 t/h の条件ではほぼ同程度のゆらぎの大きさであることが分かる。

ので、その仮定でうまく高周波側でゆらぎがキャンセルされることがあれば、重力勾配雑音を見たときに雑音が小さくなる。しかしこのキャンセルにはゆらぎの大きさのみではなく、テストマスとの距離も同一である必要があるため、難易度が高い。この状態は例えば、先行研究で提案されていたモデル 2 の状態であり、水面がある波の形を保った状態で並行移動するときに見られる (モデル 2 の推定 (図 1.8) ではこのキャンセルのため、高周波側で雑音が急激に小さくなっている)。つまり、水面がモデル 2 の場合各地点を比べた時に高いコヒーレンスが得られるはずである。しかし図 7.5 を見ると 1 Hz 以下では広い部分でコヒーレンスが高く、モデル 2 の状態を説明できるが、図 7.6 を見ると、近い部

分のみコヒーレンスが高いので、1 Hz 以上の高周波側ではモデル 1 のように波が形を保って移動している状態ではないことが分かる。

よってこの考察では、181 t/h が高周波でモデル 2 のような状況である可能性は低く、特異変動水面のゆらぎの強さや波の及ぶ範囲、位置関係によって重力勾配雑音の高周波側の大きさが決まると考える。しかし、特異変動水面のゆらぎの大きさと位置と重力勾配雑音の関係を定量的に評価することは本論文ではできなかった。よって図 7.4 からの考察として、181 t/h の条件では特異変動水面が他の条件よりも重力勾配雑音に影響しない場所にある、またはゆらぎが伝搬している範囲が狭いため 10 Hz 以上の重力勾配雑音が他の条件より小さくなっていると考える。(なお図 5.18 を見ると特異変動水面の 100 Hz 付近のゆらぎが 181 t/h の条件では小さいように見えるのでこれが重力勾配雑音の差となっている可能性がある)

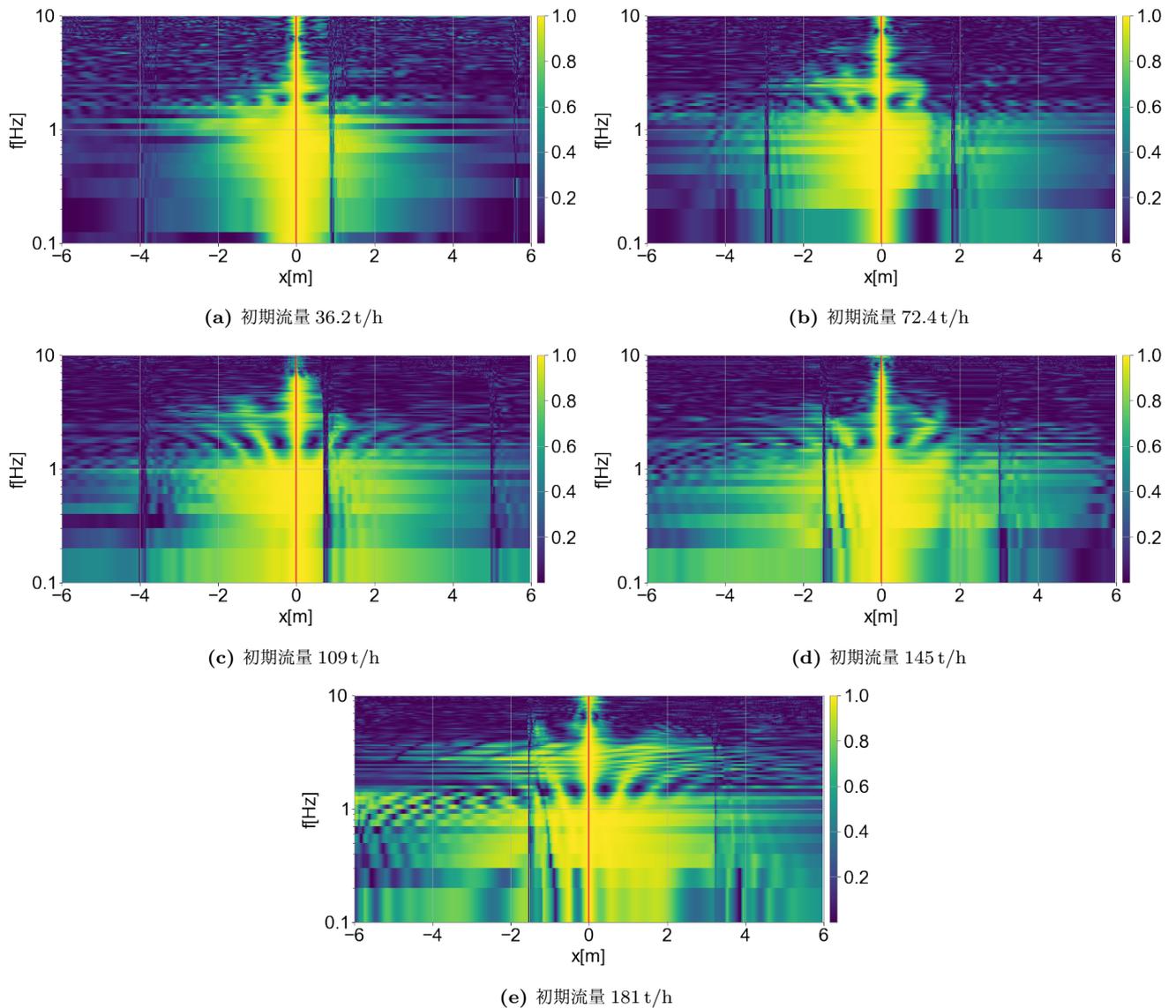


図 7.5: 赤い線の地点と他の地点の水面のゆらぎのコヒーレンス (900 秒~1000 秒)。黄色い場所が高いコヒーレンスを持つ地点である。 $x = 0$  m 地点と近傍を含む大部分の地点でコヒーレンスが高いことが分かる。

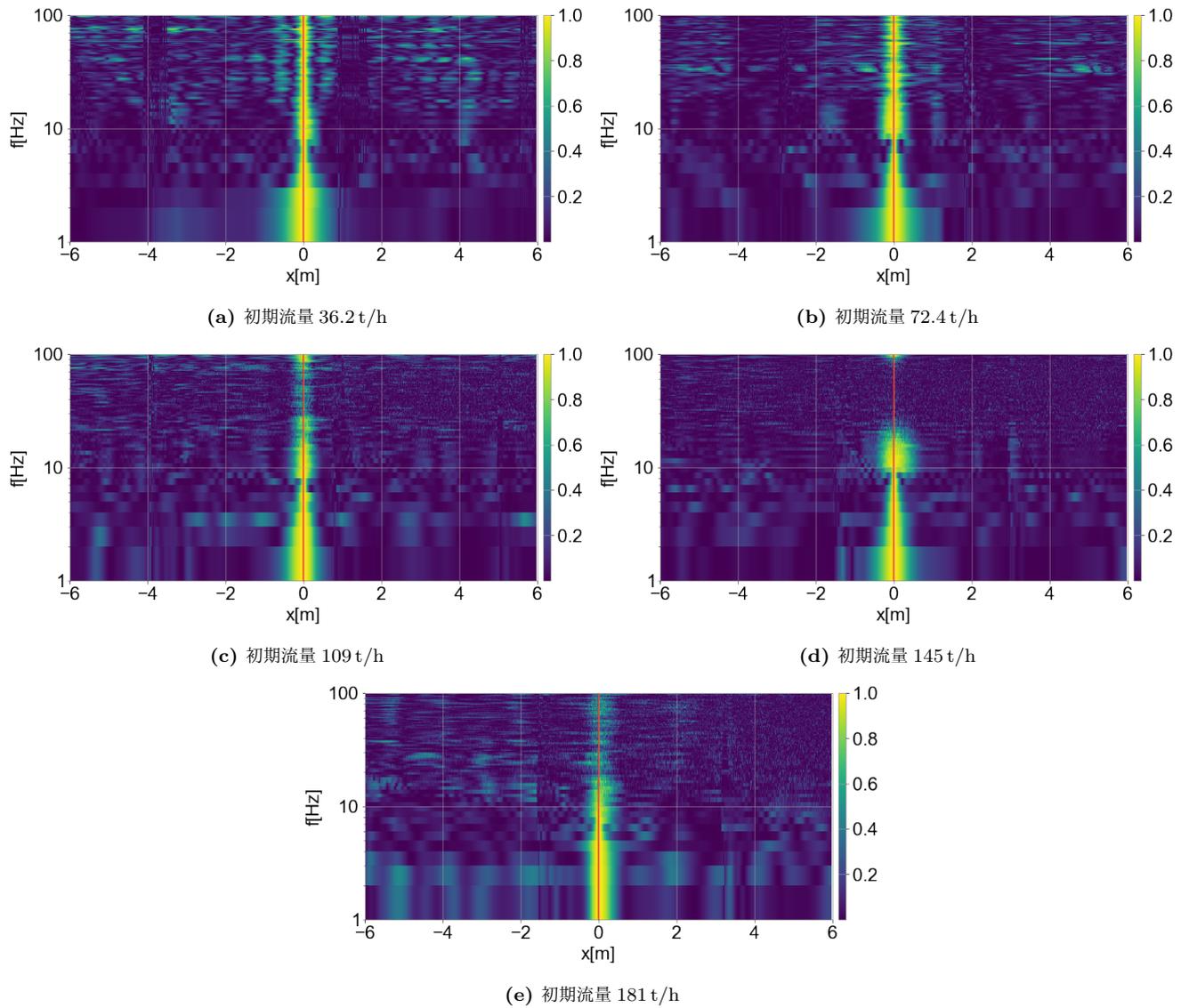


図 7.6: 赤い線の地点と他の地点の水面のゆらぎのコヒーレンス (1000 秒~1010 秒)。黄色い場所が高いコヒーレンスを持つ地点である。 $x = 0$  m 地点と近傍のみでコヒーレンスが高いことが分かる。よって 1 Hz 以上では水面の状態はモデル 2 の状態ではなく、高周波側の重力勾配雑音のキャンセルが起こらないことが考えられる。

## 7.2.2 相当粗度 (壁面粗さ) の違いによるシミュレーションの比較

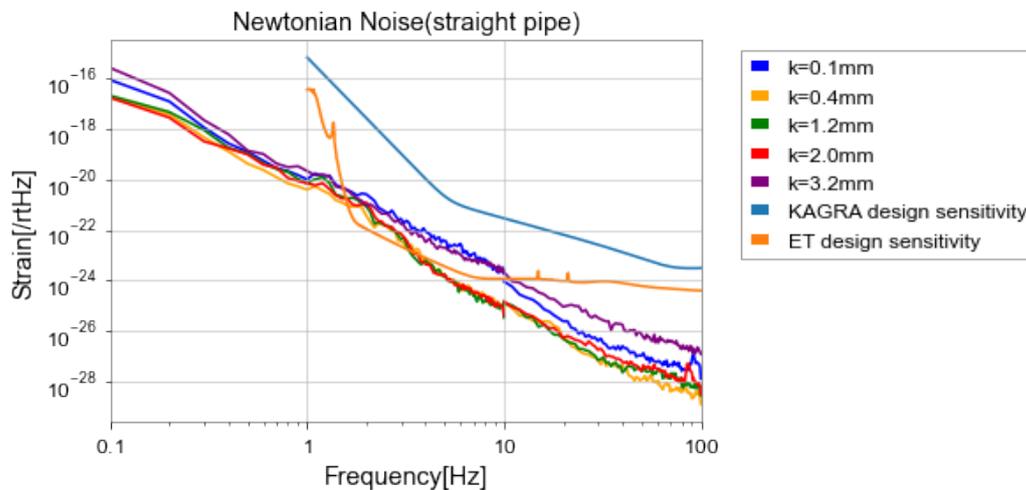
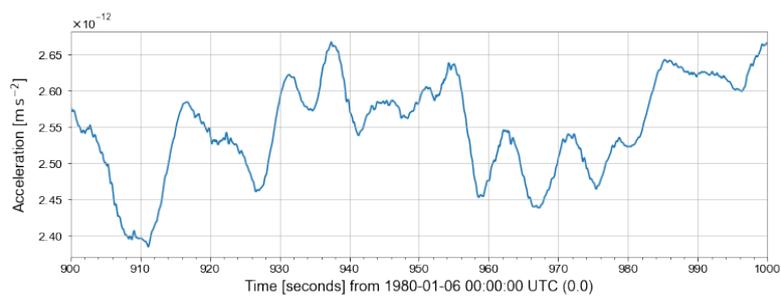


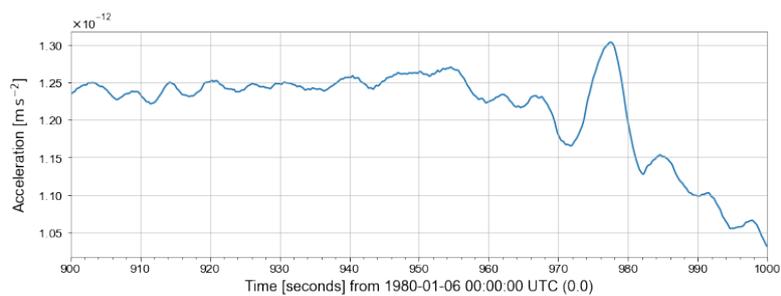
図 7.7: 直線状パイプのシミュレーションの相当粗度を変えた場合の重力勾配雑音の比較。比較したシミュレーションは表 5.1 の番号 S1、S16～S19。10 Hz 以下の雑音は 900 秒～1000 秒のサンプリング周波数 20 Hz のデータから計算されたものである。10 Hz～100 Hz の雑音は 1000 秒～1010 秒のサンプリング周波数 200 Hz のデータから計算されたものである。2 Hz の周波数から相当粗度 0.1 mm と 3.2 mm の条件で重力勾配雑音が大きいことが分かる。またこの 2 つの条件においても 10 Hz 以上で大きさの差が出ていることが分かる。

相当粗度の違いによるシミュレーションを解析した結果を示す。まず重力勾配雑音についての解析の結果を示す (図 7.7)。参考に重力勾配の影響による鏡にかかる加速度の時系列データについても載せる (図 7.8)。

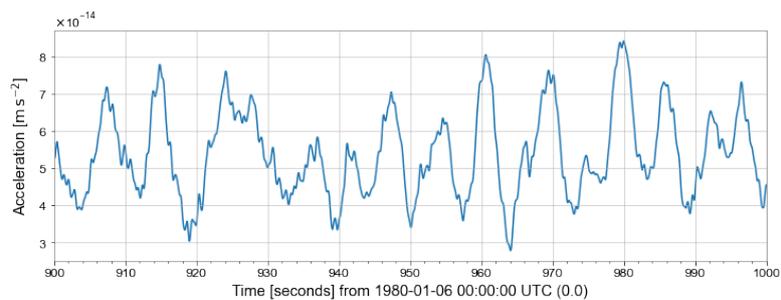
図 7.7 を見ると、本シミュレーションの条件ではどの相当粗度においても水からの重力勾配雑音は KAGRA の設計感度を越えない (1 Hz～100 Hz)。しかし、ET の設計感度を 1 Hz～10 Hz の領域で越えてしまう。さらに各条件の重力勾配雑音を比較する。まず、10 Hz までを見ると相当粗度 0.1 mm と 3.2 mm が雑音が大きく、他の条件はそれより小さくほぼ同じ大きさである。次に 10 Hz～100 Hz を見ると、10 Hz まではほぼ同じ大きさであった 0.1 mm と 3.2 mm の条件で差が現れており、0.1 mm の条件が 3.2 mm の条件に比べて雑音が小さくなっている。また、本シミュレーションの条件では相当粗度と重力勾配雑音の大きさに比例関係は見られなかった。特に相当粗度が最大の条件と最小の条件で雑音が大きく、相当粗度の大きさと相関があまりない現象が雑音の大きさに関わっている可能性がある。このように相当粗度の違いはミリメートル単位のものだが、結果の重力勾配雑音に変化をもたらすため、重要なパラメータと見ることができる。なお、結果的には本シミュレーションの相当粗度条件の範囲では KAGRA の感度と比較した場合許容できる範囲で重力勾配雑音は収まっている。よって直線状パイプの相当粗度の比較により相当粗度 0.1 mm ～ 3.2 mm の範囲では極端に重力勾配雑音が大きくなるような条件はないことが分かったので、KAGRA のパイプ配置のシミュレーションでは理論値から計算される相当粗度 3.2 mm を設定してシミュレーションを行った。



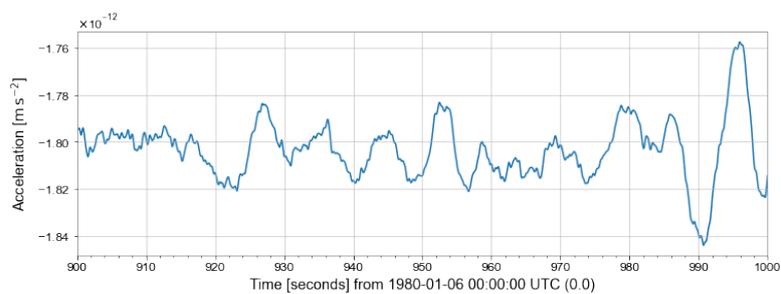
(a) 相当粗度 0.1 mm



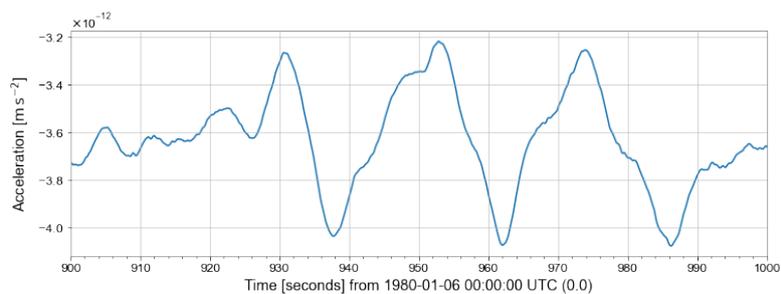
(b) 相当粗度 0.4 mm



(c) 相当粗度 1.2 mm



(d) 相当粗度 2.0 mm



(e) 相当粗度 3.2 mm

図 7.8: テストマスにかかる加速度 (重力勾配雑音) の 900 秒~1000 秒の時間変化。

### 重力勾配雑音と水面のゆらぎとの関係についての考察

初期流量の比較の時と同様に水面のゆらぎと重力勾配雑音との関係を考察する。全体的なゆらぎの傾向として初期流量の比較と同様に低周波側の水面のゆらぎが大きかった。この影響は重力勾配雑音にも現れており、重力勾配雑音においても低周波側の大きさが大きい。次に相当粗度を変えたシミュレーションにおいても特異変動水面が発生しているためこの影響を考える。図 7.9 は特異変動水面ではない  $(x, y) = (0, 0)$  地点の比較である。初期流量の比較と同様にゆらぎの大きさ (図 5.20) と位相的 (図 7.10) が他の特異変動水面ではない地点とほぼ同様であり、代表地点とできる。図 7.9 を見ると全ての条件で 20 Hz~100 Hz の周波数でほぼ同じ大きさのゆらぎを持っていることが分かる。一方、重力勾配雑音 (図 7.7) を見ると、相当粗度 3.2 mm の条件では 10 Hz 以上の周波数で他の条件との差がある。よってこの重力勾配雑音の差は図 7.9 で見たように特異変動水面ではない水面のゆらぎでは説明できないため、特異変動水面の影響で生じていると考えることができる。さらに、水面 (図 5.20) を見ると、0.1 mm と 3.2 mm の条件で特異変動水面が他の条件よりも顕著であったことからこの条件では特異変動水面の影響が重力勾配雑音の 10 Hz 以上の周波数に現れていると考えられる。また、水面の状態が高周波のキャンセルが起こるモデル 2 であるかを確認するために 1 Hz 以上のコヒーレンス図 7.11 を見た。これを見ると初期流量の比較と同様に水面の 1 Hz 以上のゆらぎのコヒーレンスは近傍のみで高く、水面はモデル 2 の状態でないことが分かる。

よって相当粗度の比較においても特異変動水面の影響が重力勾配雑音に現れていることが分かった。また、相当粗度 0.1 mm と 3.2 mm の条件で特異変動水面が多く発生し、重力勾配雑音を大きくしていることから相当粗度と特異変動水面の影響による重力勾配雑音の増大には相関が考えられない。

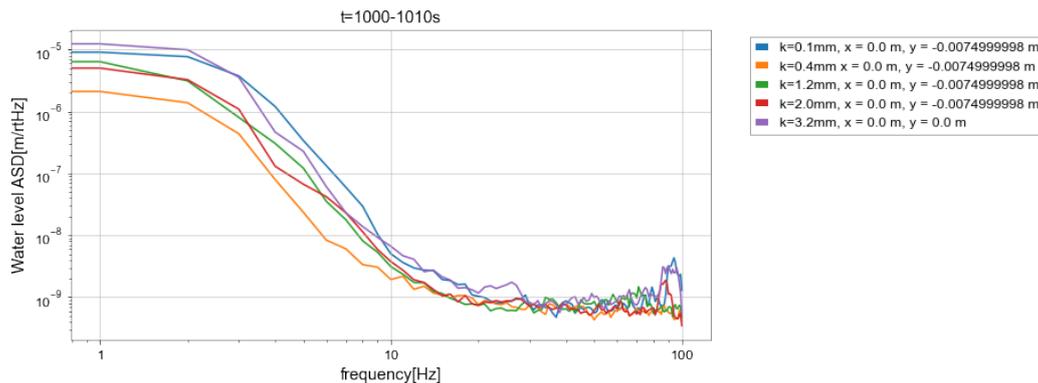


図 7.9: 各条件における  $(x,y)=(0,0)$  地点の水面のゆらぎの比較。特異変動水面ではない地点の比較として、この地点を選んだ。また、この地点は  $y = 0$  m の線上ではテストマスに最も近い位置であり、重力勾配雑音に関する影響を見る地点として最適である。1000 秒から 1010 秒の水面を解析した結果である。全ての条件で 10 Hz 以上の

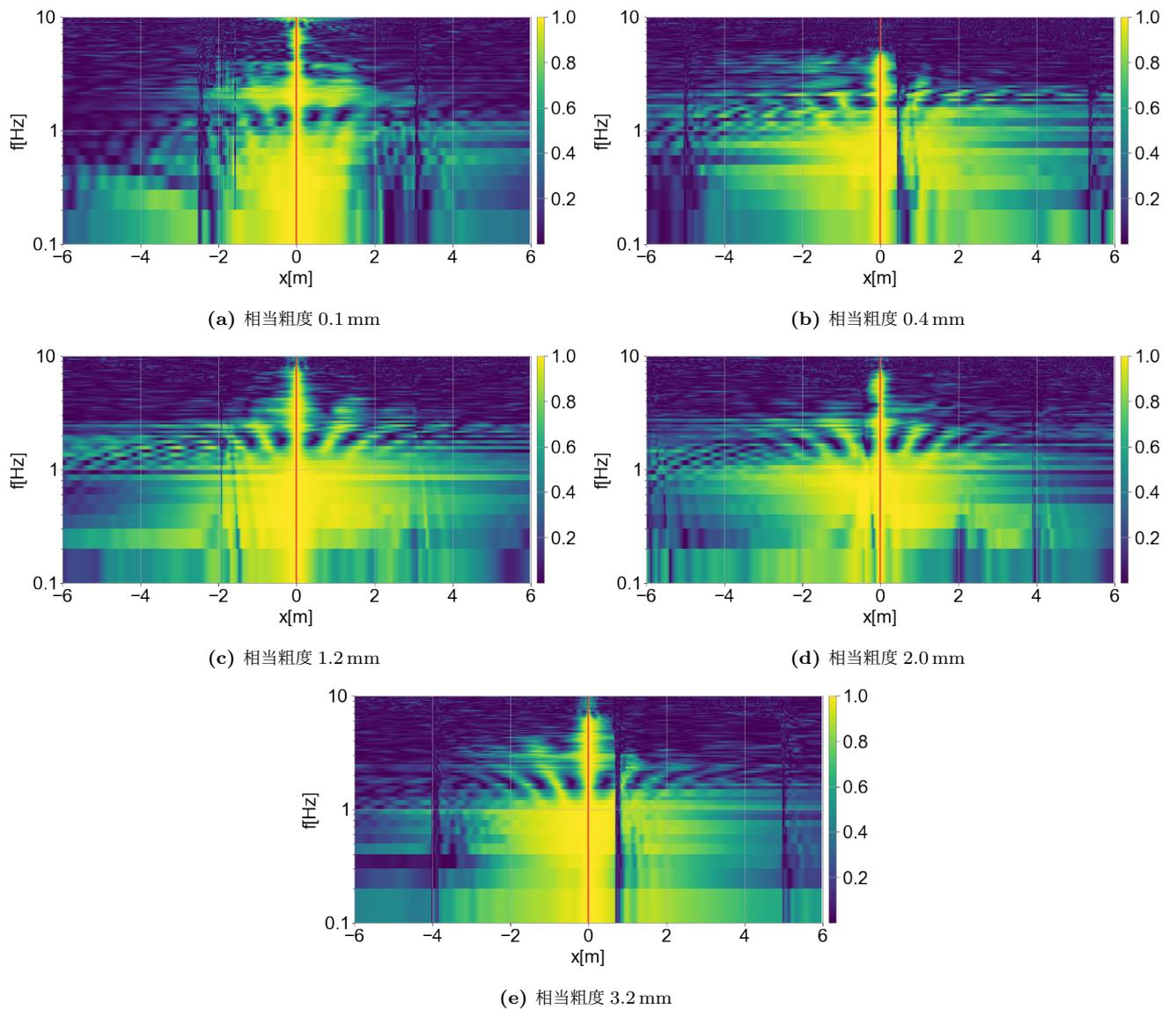


図 7.10: 赤い線の地点と他の地点の水面のゆらぎのコヒーレンス (900 秒~1000 秒)。黄色い場所が高いコヒーレンスを持つ地点である。初期流量の比較と同様に  $x = 0$  m 地点と近傍を含む大部分の地点でコヒーレンスが高いことが分かる。

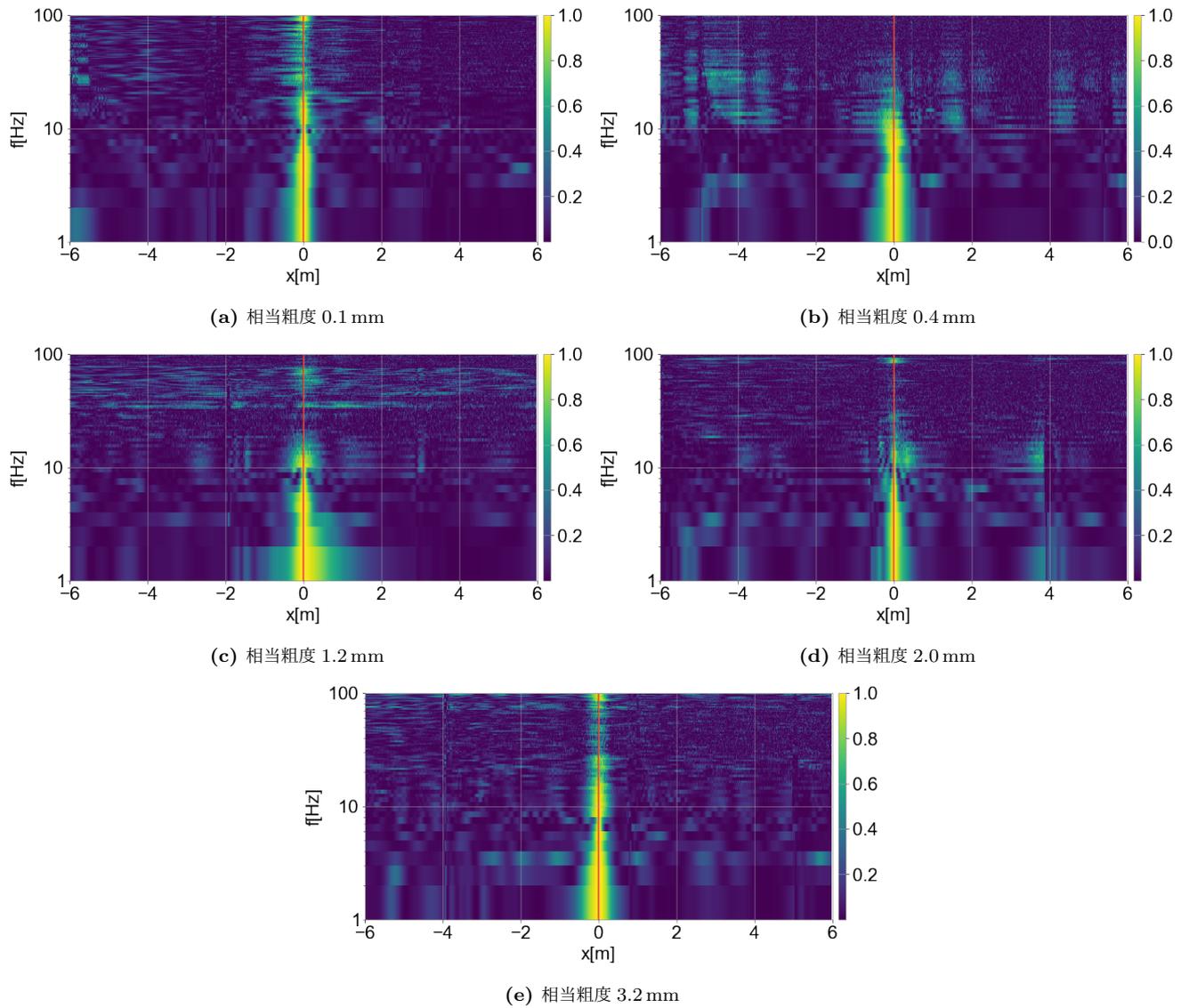


図 7.11: 赤い線の地点と他の地点の水面のゆらぎのコヒーレンス (1000 秒~1010 秒)。黄色い場所が高いコヒーレンスを持つ地点である。 $x = 0$  m 地点と近傍のみでコヒーレンスが高いことが分かる。よって相当粗度の比較においても水面はモデル 2 の状態ではないことが分かる。

## 7.2.3 傾斜の違いによるシミュレーションの比較

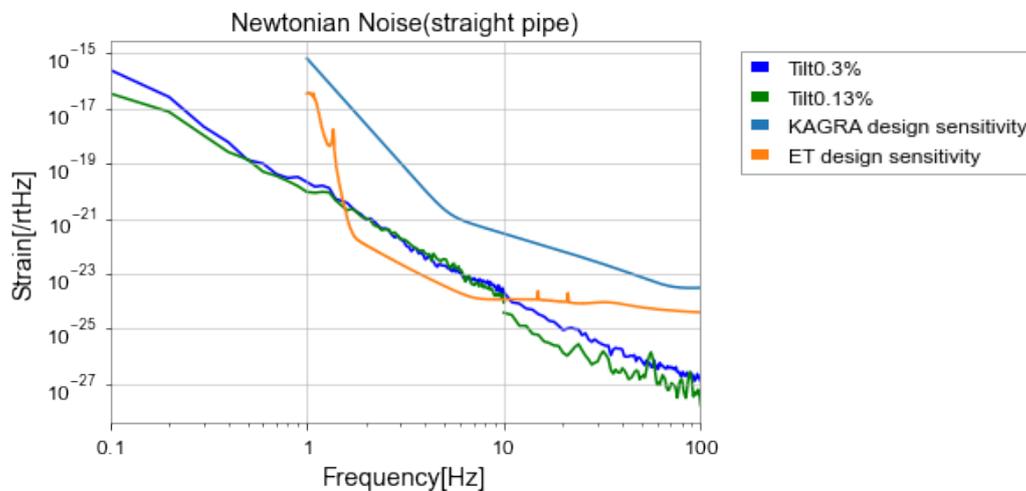


図 7.12: 直線状パイプのシミュレーションの傾斜を変えた場合の重力勾配雑音の比較。比較したシミュレーションは表 5.1 の番号 S1、S2 である。10 Hz 以下の雑音は 900 秒～1000 秒のサンプリング周波数 20 Hz のデータから計算されたものである。10 Hz～100 Hz の雑音は 1000 秒～1010 秒のサンプリング周波数 200 Hz のデータから計算されたものである。低周波側から 10 Hz までの重力勾配雑音の大きさはどちらの傾斜においてもほぼ同じであるが 10 Hz 以上の大きさに差が生じていることが分かる。

傾斜の違いによるシミュレーションを解析して得られた重力勾配雑音についての解析の結果を示す (図 7.12、図 7.13)。図 7.12 を見ると、傾き 0.3 %、0.13 % いずれにおいても重力勾配雑音は KAGRA の設計感度を越えない。しかし、ET の設計感度を 1 Hz～10 Hz の帯域で越えてしまう。また、10 Hz までを見ると傾斜によっては雑音の大きさに大きな差はないことが分かる。10 Hz 以上の帯域を見ると差が生じているが、傾斜 0.13 % の条件では 10 Hz での大きさが 100 秒間と 10 秒間のデータで異なっており、100 秒間でとらえられていた水面のゆらぎが 10 秒間のデータでうまくとらえられていなかった可能性が考えられる。

## 重力勾配雑音と水面のゆらぎとの関係についての考察

傾斜 0.13 % の条件で重力勾配雑音は 10 Hz では断続的ではあるが水面のゆらぎ図 5.16、図 5.22 では 10 Hz 以上の重力勾配雑音に影響すると考えられる特異変動水面の個数が傾斜 0.3 % と 0.13 % で異なっており、この影響で重力勾配雑音の大きさが異なっていると考えられる。また、高周波側でゆらぎがキャンセルされ重力勾配雑音が小さくなるモデル 1 の状態であるかをコヒーレンスを見て確認した (図 7.14)。これを見ると、他の直線状のパイプと同様に水面はモデル 2 の状態でなく高周波側でゆらぎがキャンセルされ重力勾配雑音が小さくなることは考えられない。

よって傾斜の違いのシミュレーションにおいても特異変動水面が重力勾配雑音に影響を与え特異変動水面が少ない傾斜 0.13 % の条件のほうが重力勾配雑音の 10 Hz 以上の大きさが小さくなることが分かった。また、ここまでの直線状のパイプの考察により特異変動水面が重力勾配雑音の 10 Hz 以上の部分の大きさに影響を与えることが分かった。よって 10 Hz 以上の重力勾配雑音を低減するには特異変動水面を発生させない、ゆらぎを小さくする、テストマスから離すなどの措置が必要であることが分かる。しかし、5 章でも述べた通り本論文で行ったシミュレーションの条件の比較から特異変動水面のコントロールは難しく、重力勾配雑音の低減も難しいことが分かる。

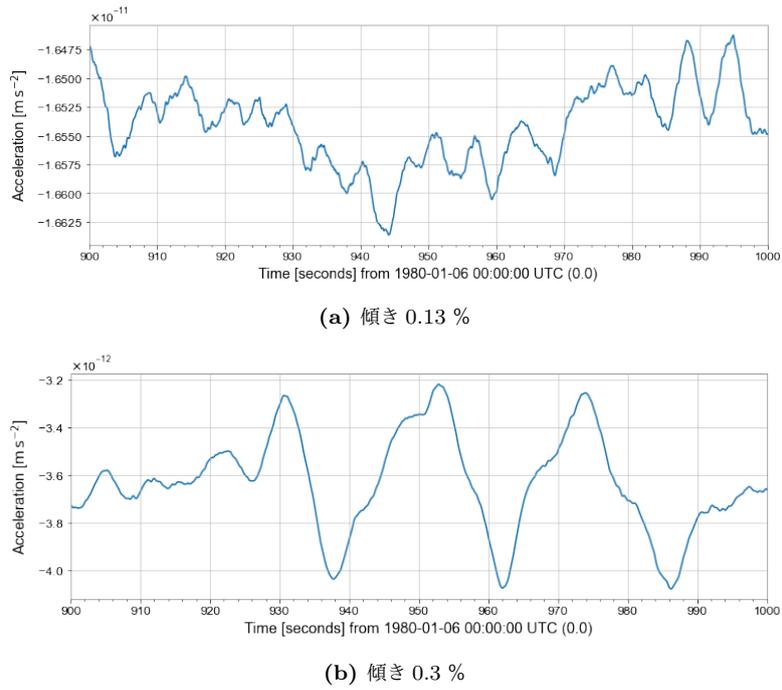


図 7.13: テストマスにかかる加速度 (重力勾配雑音) の 900 秒~1000 秒の時間変化。

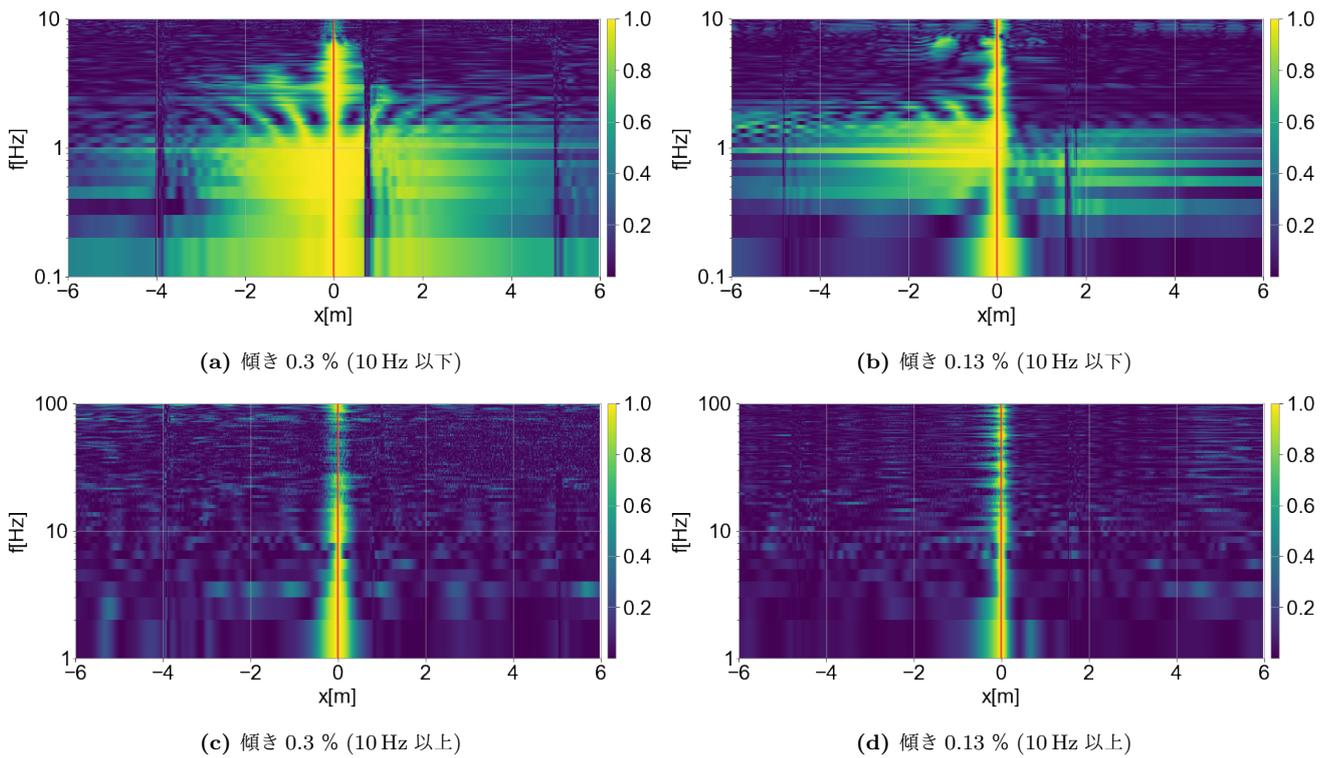


図 7.14: 赤い線の地点と他の地点の水面のゆらぎのコヒーレンス。黄色い場所が高いコヒーレンスを持つ地点である。10 Hz 以下 (上段) では  $x = 0$  m 地点と近傍を含む大部分の地点でコヒーレンスが高いことが分かる。一方 10 Hz 以上 (下段) では  $x = 0$  m 地点と近傍のみの地点でコヒーレンスが高いことが分かる。よって水面の状態はモデル 2 ではなく高周波がキャンセルされ重力勾配雑音の高周波側の大きさが小さくなることはない。

### 7.3 KAGRA のパイプ配置 (Y-end のパイプ) における地下水のシミュレーション

Y-end のパイプを再現したパイプのシミュレーションの結果について説明する。3つの初期流量の条件の重力勾配雑音は図 7.15 のようになった。これを見るとどの条件においても KAGRA の設計感度を越えておらず KAGRA の感度に影響を及ぼさないことが分かる。この Y-end のパイプを再現したパイプは現実の系に近く、この結果は実際の Y-end の排水パイプ (中央排水管) を通る水からの重力勾配雑音に近いものであると言える。よって KAGRA の Y-end では本論文のシミュレーションの範囲では水の重力勾配雑音は感度に影響を与えないと言える。さらに参考として鏡にかかる加速度 (重力勾配雑音) の時間変化を載せる (図 7.16、図 7.17)。

条件ごとの比較をすると初期流量 181t/h の条件の雑音が最も大きいことが分かる。また、36.2t/h と 109t/h の条件では雑音の大きさは 10Hz 以上ではほぼ同程度である。さらに 109t/h、181t/h の条件では 1Hz~10Hz でいくつかのピークが見られる。さらに図 7.17 を見ると、初期流量 109t/h と 181t/h の条件の 1Hz~10Hz ピークの構造が見える。

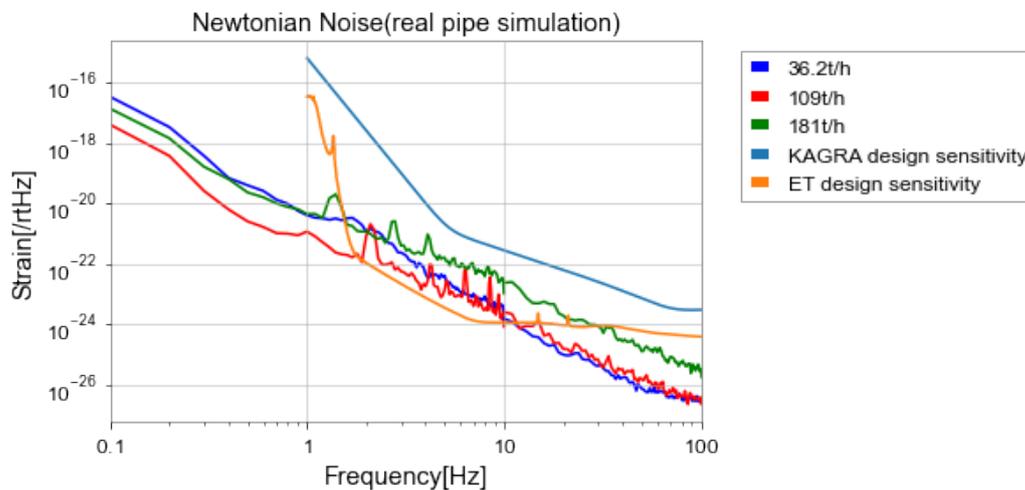
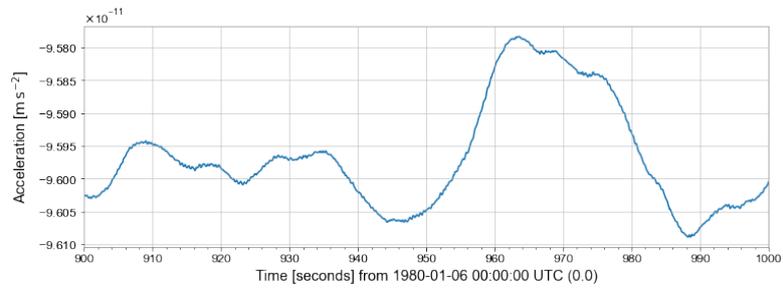
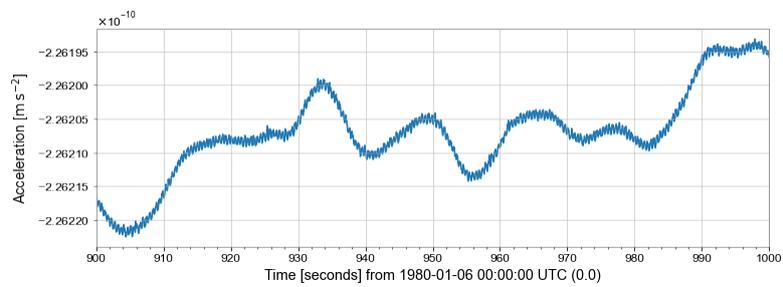


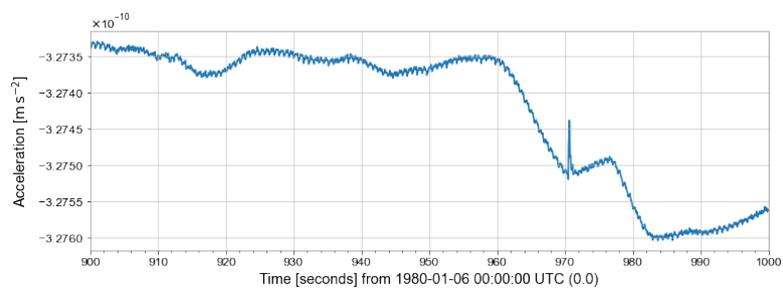
図 7.15: 10 Hz 以下の雑音は 900 秒~1000 秒のサンプリング周波数 20 Hz のデータから計算されたものである。シミュレーション番号は K1、K2、K3 である。10 Hz~100 Hz の雑音は 1000 秒~1010 秒のサンプリング周波数 200 Hz のデータから計算されたものである。重力勾配雑音を比べると低周波側から 10 Hz までの重力勾配雑音の大きさは初期流量 36.2 t/h と 181 t/h の条件で大きく、10 Hz 以上では 初期流量 181 t/h の条件で大きくなっている。また、初期流量 109 t/h と 181 t/h の条件では 1 Hz~10 Hz でピークが存在している。



(a) 初期流量 36.2 t/h

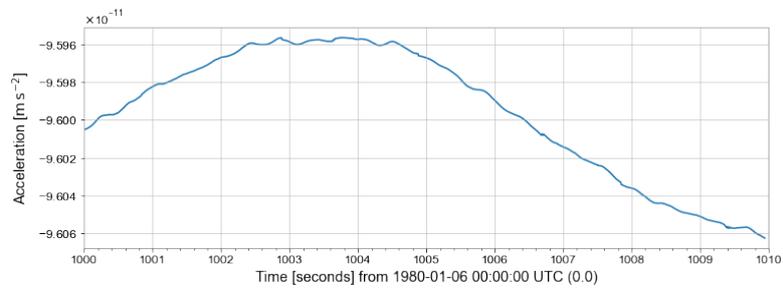


(b) 初期流量 109 t/h

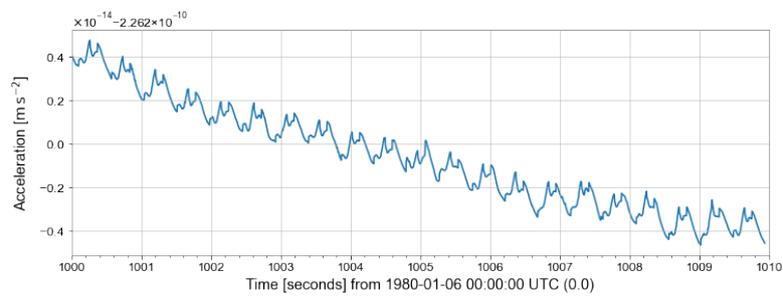


(c) 初期流量 181 t/h

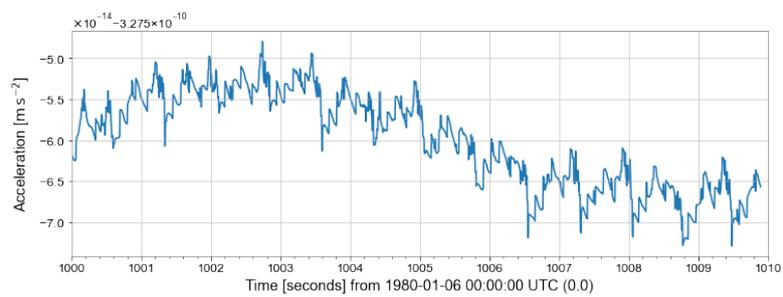
図 7.16: テストマスにかかる加速度 (重力勾配雑音) の 900 秒~1000 秒の時間変化。



(a) 初期流量 36.2 t/h



(b) 初期流量 109 t/h



(c) 初期流量 181 t/h

図 7.17: テストマスにかかる加速度 (重力勾配雑音) の 1000 秒~1010 秒の時間変化。1 Hz~2 Hz の構造が見えていることが分かる。

### 重力勾配雑音と水面のゆらぎとの関係についての考察

図 7.15 で見える 1 Hz~10 Hz にあるピークは 6 章における水面の考察において、水面のゆらぎでみられたピークと一致する。6 章における水面の考察ではこのピークは straight2 の曲がり角の水面で発生しており、パイプの壁面からのゆらぎではなく曲がり角で発生した 1 Hz~2 Hz のゆらぎが直線部分である straight2 に伝搬しており水面全体にこのピークが見えているという考察をした。よってこのピークと対応する周波数の重力勾配雑音でピークが見えていることから重力勾配雑音の 1 Hz~10 Hz にあるピークを小さくするためには曲がり角の水のゆらぎを低減することが必要である。本シミュレーションの初期流量の条件ではこのピークは KAGRA の感度を制限しない大きさだが Y-end のパイプを再現したパイプにおいてこのピークが顕著になっているため、実際の KAGRA のパイプにおいてもこのピークが大きくなるか注意する必要がある。なお、本論文のシミュレーションでは初期流量 36.2 t/h ではこのピークのゆらぎが抑えられており (図 6.3)、重力勾配雑音においてもピークが見えていないため初期流量が少ない条件ではこのピークの大きさが抑えられると考えられる。

次に直情状パイプのシミュレーションで見えていた特異変動水面だが、Y-end のパイプの構造では 10 Hz 以上でゆらぎが大きい特異変動水面は見えていない (図 6.4)。よって AGRA のパイプ配置 (Y-end のパイプ) における地下水のシミュレーションでは重力勾配雑音の 10 Hz の部分では特異変動水面による影響を考える必要がない。なお、初期流量 181 t/h の重力勾配雑音のが 3 Hz 以上で他の条件より大きい原因は、特異変動水面ではないが 26 Hz で見えていた水面のゆらぎや、straight2 以外の水面が原因である可能性がある。また、モデル 2 の状態である時の高周波側のキャンセルだがここでも straight2 のコヒーレンスを見て判断をした (図 7.18)。これを見ると直線状パイプの時と同様に 1 Hz より大きい周波数のゆらぎのコヒーレンスは対象地点と近傍のみであり、高周波側でキャンセルするモデル 2 のような状態ではないことが分かった。よって Y-end におけるパイプにおいても水面の状態はモデル 2 ではなく、重力勾配雑音においても高周波側のゆらぎのキャンセルによる雑音の低下は考えられない。

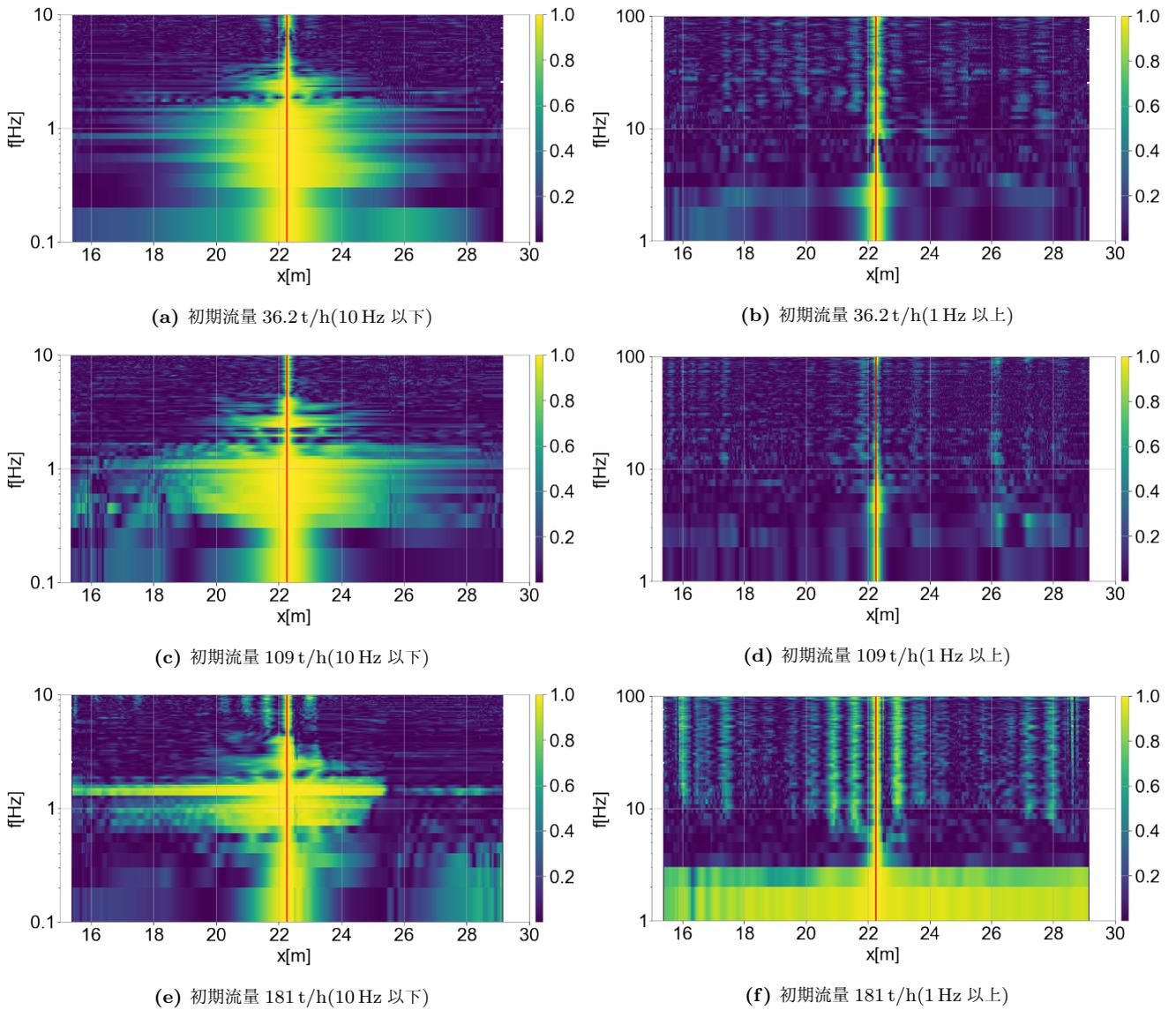


図 7.18: 赤い線の地点 ( $x = 22\text{ m}$ ) と straight2 の他の地点の水面のゆらぎのコヒーレンス。黄色い場所が高いコヒーレンスを持つ地点である。10 Hz 以上では水面の状態はコヒーレンスが近傍のみで高くモデル 2 の状態ではなく、高周波側の重力勾配雑音のキャンセルが起こらないことが考えられる。なお 181 t/h では 10 Hz 以上でコヒーレンスの断続的な構造が見えるが、断続的であるため波が伝搬している状態ではないと考えられる。

## 第 8 章

# 結論

### 8.1 本論文における結論

重力勾配雑音の解析結果を見ると、本論文で行ったシミュレーションの範囲ではパイプの水からの重力勾配雑音は KAGRA の設計感度を制限しないことが分かる。本研究は KAGRA の Y-end におけるパイプ内の水が引き起こす重力勾配雑音を対象としているが、Y-end のパイプを再現したパイプでも KAGRA の設計感度を制限しないため、結論として本論文のシミュレーションで設定した流量では KAGRA においてパイプの水からの重力勾配雑音は KAGRA に影響を及ぼさないとと言える。

さらにシミュレーションより得られた重力勾配雑音の解析結果を見ると本論文における条件の範囲では直線状のパイプのシミュレーションでは初期流量 181 t/h の条件、傾斜が緩やかな条件、KAGRA のパイプ配置 (Y-end のパイプの配置) における地下水のシミュレーションでは初期流量が 36.2 t/h、109 t/h の条件で重力勾配雑音は小さいことが分かった。現実の KAGRA のパイプを考えると Y-end のパイプを再現したパイプの結果を参考に初期流量は少ないほうが重力勾配雑音は小さいと言える。

仮定として ET の鏡が KAGRA の Y-end と同様の状況であった場合本論文のシミュレーションの範囲では 1 Hz～10 Hz の帯域で ET の設計感度を制限してしまうという結果が出ている。そのため、ET の建設においては地下水を排出する際に鏡に近づけない設計をする必要性が提案される。また、本論文執筆段階では建設地が決定されていないが、地下水が湧かない場所を選ぶという提案もできる。

#### 8.1.1 水面の状態と重力勾配雑音の大きさの関係

次に水面の状態に触れて結論としてまとめる。まず、直線状パイプにおいていくつかの地点で特異変動水面が出現することが分かった。仮定として特異変動水面は「跳水」であるという予測をしてフルード数を調べた結果、跳水ではない別の原因で特異変動水面が現れている地点が多いことが分かった。本論文では特異変動水面のゆらぎの原因については断定できなかった。さらに、特異変動水面のゆらぎ調べたところ 10 Hz～100 Hz でもゆらぎが大きいことが分かった。このゆらぎは重力勾配雑音で見た時にも 10 Hz 以上の部分で影響を与えており 10 Hz 以上の重力勾配雑音を考える時に特異変動水面は重要となる現象であることが分かった。また、フルード数的には本論文のシミュレーションは常流の状態が多かったため常流の状態から受ける重力勾配雑音を調べたとも言える。なお、KAGRA のパイプ配置における地下水のシミュレーションでは straight2 が常流であったことから現実の KAGRA の Y-end の水流も常流である可能性が高い。

さらに KAGRA のパイプ配置における地下水のシミュレーションでは重力勾配雑音と水面において 1 Hz～10 Hz でいくつかのピークが見られており、本論文のシミュレーションの条件では KAGRA の感度を制限しないものの、条件

が違えばゆらぎが大きくなる可能性があり、懸念される。この重力勾配雑音において 1 Hz～10 Hz で見えていたピークは曲がり角や straight2 の水面全体で見えており、これらの水面からの影響であることが分かった。また、水面におけるピークとなるゆらぎが発生する原因は曲がり角で 1 Hz～2 Hz をピークとするゆらぎが発生しその後の水面の伝搬しているということも分かっている。そのため、現実の KAGRA の重力勾配雑音においては曲がり角の影響を少なくすることで 1 Hz～10 Hz におけるピークを低減することができると結論付けられる。

### 8.1.2 重力勾配雑音の低減

#### 特異変動水面

本論文における議論では水面の様子を調べ重力勾配雑音との関係を考察した。その結果、直線状のパイプでは重力勾配雑音の 10 Hz 以上で特異変動水面が影響している可能性が高いという結論が得られた。KAGRA は 10 Hz 以上の帯域で感度がよく、10 Hz 以上の雑音を低減することが重要となる。そのため 10 Hz 以上の重力勾配雑音を低減するには特異変動水面をテストマスから遠ざける、無くす、ゆらぎを小さくするといったことが必要になる。さらに特異変動水面は一部水理学から考えられる跳水である可能性があるが、他の多くの地点では跳水以外の要因でゆらぎが大きかった。

直線状パイプの比較により、初期流量、相当粗度、傾斜を変化させると特異変動水面の位置や個数が大きく変わることが分かった。これは特異変動水面を流量などによりコントロールすることは難しいということを示す。よって意図的にテストマスから遠ざけることは難しいということになる。次に無くす方法だが、跳水であった場合は跳水を無くすことは多くの水路で行われており\*1、例えば傾斜をつけることで水路全体を射流にして回避する方法などがある。しかし、KAGRA の Y-end の中央排水管は床下に埋まっており、工事が難しいという状況でありパイプ自体に何らかの変化を加えることは難しい。さらに、水量をコントロールして特異変動水面を無くすことは前述の通り難しく、意図的に無くすという方法は取れない。次に、ゆらぎを小さくする方法だが、特異変動水面のコントロールが難しいことに加えて現実のパイプは途中で曲がり角があるなど、ゆらぎの大きさに関わる不確定要素が多く難しいと考えられる。なお、次に KAGRA のパイプ配置における地下水のシミュレーションでは特異変動水面は発生しておらず、本論文のシミュレーションで設定した初期流量の範囲では現実の KAGRA のパイプでも発生しない可能性が考えられる。よって現実の KAGRA のパイプではすでに特異変動水面の発生が回避されており、10 Hz 以上の重力勾配雑音が抑えられていると見方もできる。ただし、他の初期流量の条件で Y-end のパイプの構造で特異変動水面が発生する可能性は否定できず、引き続き現実の KAGRA のパイプにおいても特異変動水面の発生に注意する必要がある。

よって本論文における 10 Hz 以上の重力勾配雑音を低減する方法としては特異変動水面自体のコントロールは難しいのでパイプ自体の距離をテストマスから離す、またはパイプ内の水を満水状態にして水のゆらぎを無くすといった方法がある。また、研究が進み Y-end のパイプにおける、跳水や水面のゆらぎの大きさと初期流量や流速、フルード数などの関係が判明すれば水をコントロールすることで特異変動水面をコントロールし重力勾配雑音を低減することができる可能性がある。

#### Y-end のパイプにおける 1 Hz～2 Hz のピーク

KAGRA のパイプ配置における地下水のシミュレーションでは曲がり角で 1 Hz～2 Hz をピークとするゆらぎが発生し他の水面に広く分布することが分かった。そしてそのピークは重力勾配雑音にも影響を与え、重力勾配雑音においても 1 Hz～2 Hz のピークが見られた。このピークの大きさは本論文のシミュレーションの範囲では KAGRA の感度を制限しないが、他の条件では大きくなる可能性があり、低減する必要がある。曲がり角で発生していることが判明して

\*1 跳水は水路へのダメージが大きいため、一般的な水路においても回避されるようにすることがある

いるのでまず、曲がり角を無くすといった方法が考えられる。しかし、前述した通り Y-end のパイプは地下に埋まって入り、曲がり角を無くすという手段は大がかりなものになる可能性があり推奨されない。次に、水量をコントロールして低減する方法であるが本論文のシミュレーションでは初期流量 36.2t/h において 1Hz~2Hz をピークするゆらぎの大きさが小さくなっておりパイプに流れる水量を少なくするのが良いと考えられる。

### 8.1.3 先行研究における理論との比較

1.4.1 節にて先行研究において水面のゆらぎのモデルとして、水面が各地点でゆらぎ空間方向への伝搬を考えないモデル 1 と水面が形を保って移動するモデル 2 の 2 つのモデルが提案されていることを紹介した ([13])。モデル 1 では空間方向への伝搬を考えないので、各地点同士の水面のゆらぎのコヒーレンスが低いと考えられ、モデル 2 では水面が形を保って移動しているので、水面のゆらぎのコヒーレンスが高いと考えられる。ここで直線状パイプの初期流量の比較のコヒーレンス (図 7.5、図 7.6、図 7.10、図 7.11、図 7.14) を見ると、1Hz 以下では  $x = 0\text{m}$  地点と他の多くの地点でコヒーレンスが高いが、1Hz 以上では近傍のみコヒーレンスが高いという結果が得られた。これは 1Hz 以下の水面のゆらぎはモデル 2 に近く、1Hz 以上の水面のゆらぎはモデル 1 に近いと言える。このことから図 1.8 の各モデルの重力勾配雑音の見積を参考にすると 1Hz 以下の重力勾配雑音ではモデル 2 のようになるが、1Hz 以上ではモデル 1 のようになり、高周波側ではモデル 2 の見積どおり急激に重力勾配雑音が高周波側で小さくならないと言える。実際に、7 における重力勾配雑音の結果においても高周波側でモデル 2 のように急激に下がってないことが分かる。また、図 7.18 を見ると Y-end のパイプの水面においても同様に、高周波側ではモデル 2 ではなく高周波側で重力勾配雑音が急激に下がらないと言える。

## 8.2 本研究の将来的な展望

ここでは本論文の結果や現在の研究の状況を踏まえた、将来的な展望についてまとめる。また、本研究の改善できる点についても記載する。

まず、シミュレーションについては本論文で設計したパイプでは実際のパイプには存在する蛇腹部分が再現されていないことである。図 2.7 を見ると、実際のパイプは蛇腹になっている (パイプ内部の蛇腹は浅い凹凸である)。本シミュレーションでは内側の蛇腹の凹凸が浅いこともあり、その影響を入れずにパイプを設計したがより正確なシミュレーションを行うためにはこの蛇腹の構造を入れる必要がある。方法としては蛇腹の構造を測り、それを CAD ソフト上の設計で再現する方法が考えられる。

次に、KAGRA のパイプ配置における地下水のシミュレーションにおいて他の様々な条件でシミュレーションをすることだ。本論文では初期流量を 3 条件で比較するのみであったが、さらに多くの初期流量の条件でシミュレーションをすることで様々な流量を取る現実の Y-end におけるパイプの重力勾配雑音を予測することができる。さらに、現実の流量は一定でない場合があるので、時間により流量を変化させてシミュレーションを行うと重力勾配雑音の予測の幅が広がると考えられる。

さらに、シミュレーションの条件について触れると、本論文におけるシミュレーションは常流の状態のものが大半であった。そのため射流の状態の水面の様子は未解明なことが多く、その時に重力勾配雑音の特徴なども捉えられていない。Y-end を再現したパイプでは流れが常流であったため、現実の水面も基本的に常流であることが考えられるが、重力勾配雑音と水面の関係を探るためには射流や跳水が発生した場合の関係をみるのが重要であると考えられる。よって意図的に射流の状態を作り出し、その時の重力勾配雑音を解析することが重要となる<sup>\*2</sup>。本論文のシミュレーションで

<sup>\*2</sup> 本論文におけるシミュレーションを行う前に、間違った設定 (パイプ壁面に滑り度を追加していた) のシミュレーションを行っていた。そのシミュレーションではパイプ上流部分で射流下流部分で常流となっており、その間で跳水が発生していた。その水面では常流部分で低周波側

はテストマスとの距離が近い中央排水管を流れる水を対象にシミュレーションを行った。しかし、Y-endにはテストマスとの距離は遠いが強制排水管のパイプもあり、水が流れている。そのため強制排水管のパイプ内の水もシミュレーションを行い重力勾配雑音を計算することでより実際の重力勾配雑音に近づけると考えられる。これは、例えばこのシミュレーションはパイプ1本ごとに行い、プログラム上で座標を調整することで行える。なお、本論文で行ったシミュレーションの条件ではテストマスと中央排水管の位置関係にあるとき重力勾配雑音はKAGRAの感度を制限しないので、さらに遠い位置にある強制排水管の影響は小さいと考えられる。

次にFlow-3Dを動かすPCのスペックをあげて、さらにシミュレーションの精度精度をあげる。という改善が考えられる。本論文では

— PC のスペック —

プロセッサ Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz 4.00 GHz  
実装 RAM 64.0 GB (63.7 GB 使用可能)  
システムの種類 64 ビット オペレーティング システム、x64 ベース プロセッサ  
OS Windows10 education  
SSD 7TB × 2  
GPU(CPU 内蔵)

というスペックのPCでFlow-3Dを動かした。本論文のシミュレーションではシミュレーション時間などで精度が制限されたため例えばCPUをさらに性能の良いものに変えるなどの改善が考えられる。

最後に本研究の新たな課題として特異変動水面という大きな水面のゆらぎの回避や、他の大きなゆらぎを産み出す現象の回避さらにはETにおける地下水の影響への考察が挙げられる。特異変動水面の回避は前述した通り、水面への考察が深まれば特異変動水面をコントロールすることが可能になる可能性がある。他の大きなゆらぎとは外的要因による突発的な水面のゆらぎのことであり、例えば水量が突然変化し水面がゆらいだり、「石」などが水面に落ちゆれることなどである。本論文のシミュレーションは水の流入以外の外的要因が全くない理想的な条件で行ったが様々な場合を考慮することは水の重力勾配雑音の研究において重要だと考える。本論文ではKAGRAにおける排水パイプのシミュレーションを行ったがKAGRAと同じく地下に建設されるETにおいて水の排水を行う可能性がある。KAGRAより重力勾配雑音の影響を受けやすいETで水の重力勾配雑音の研究を行うことは意義があるものだと考えられる。

---

のゆらぎが大きく、射流では全体的にゆらぎが小さく、跳水部分では10Hzまで全てのゆらぎが大きかった。このシミュレーションは偶然この3状態が観察され、かつ間違った設定であったため、本論文には記載していないが、これからの課題として意図的にこのような状態をつくり、特にゆらぎが大きい跳水部分などからの重力勾配雑音への影響を調べるということが重要になると考えられる。

## 付録 A

# 長方形断面の水路におけるフルード数と跳水の形について

([18]P.188、図 5-3) に記載される、長方形断面の水路における跳水開始地点のフルード数  $F_{r1}$  と跳水の形について紹介する。

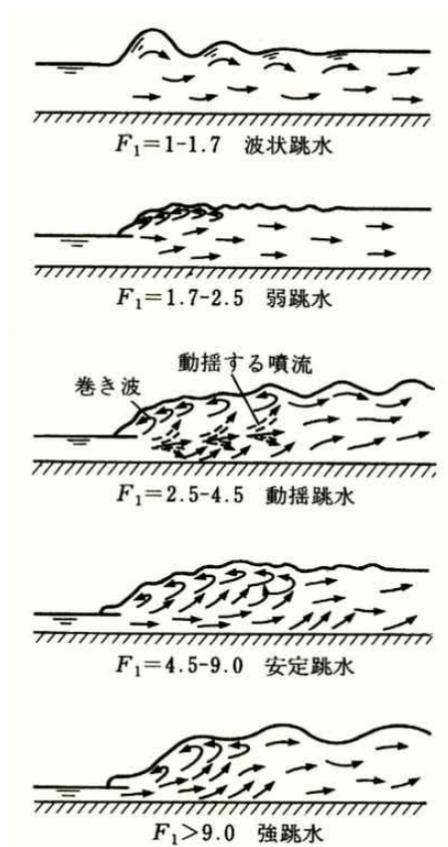


図 A.1: ([18]P.188、図 5-3) に記載される、跳水開始地点のフルード数  $F_{r1}$  と跳水の形の関係  $F_{r1}$  が 1 以上で跳水が発生し、その値によって跳水の形が異なる。跳水は流体の渦で構成されており、水面のゆらぎが大きくなる。

## 付録 B

# 平均速度公式の定数

[16] に記載される、粗度係数の物質ごとの値を記す。

表 B.1: ガンギエ-クッターまたはマニングの公式の  $n$ [16]

水路の種類	$n$
閉管路	
黄銅管	0.009~0.013
鋳鉄管	0.011~0.015
びょう接銅管	0.013~0.017
純セメント平滑面	0.010~0.013
コンクリート管	0.012~0.016
人工水路	
滑らかな木材	0.010~0.014
コンクリート巻	0.012~0.018
切石モルタル積	0.013~0.017
粗石モルタル積	0.017~0.030
粗石空積	0.025~0.035
土の開さく水路、直行線状で等断面	0.017~0.025
土の開さく水路、蛇行した鈍流	0.023~0.030
岩盤に開さくした水路、滑らかな場合	0.025~0.035
岩盤に開さくした水路、粗な場合	0.035~0.045
自然河川	
線形、断面とも規則正しく、水深が大きい	0.025~0.033
同上で河川がれき、草岸のもの	0.030~0.040
蛇行していて淵瀬のあるもの	0.033~0.045
蛇行していて、水深が小さいもの	0.040~0.055
水草が多いもの	0.050~0.080

## 付録 C

# Flow-3D の出力を解析するために作成したプログラム

Flow-3D の出力を解析するために作成した Python3 によるプログラムを載せる。プログラムは Jupyter Notebook 上で動くように作成した。そのため元のコードが .ipybn であり、In[] が挿入されているが、コードを貼る都合で .py 形式に直した際のものであるのでここは無視して良い。ここで紹介するプログラムは解析用に複数の関数を独自に作成し、解析するものによって使う関数や記載される文字列を適宜変更する必要がある。重力勾配雑音以外にも水面を解析する部分も記載してあるが、基本的には自ら解析したいものに合わせてプログラム前半で作った関数を使用する。ここでは読み込むファイル名なども記載しているが、例である。

Code C.1: プログラム全文

```
1  #!/usr/bin/env python
2  # coding: utf-8
3
4  # In[1]:
5
6
7  import numpy as np #をインポートnumpy行列()
8  from numpy import pi
9  from scipy import integrate #の積分scipy
10 import matplotlib.pyplot as plt
11 import matplotlib.animation as animation
12 from matplotlib import cm
13 from matplotlib import colors
14 from gwpy.timeseries import TimeSeries
15 from gwpy.frequencyseries import FrequencySeries
16 from gwpy.plot import Plot
17 from tqdm import tqdm
18 import copy
19 import itertools
20 import glob
21 import re
22 import pickle
23
24
25 # In[2]:
```

```
26
27
28 def atoi(text):
29     return int(text) if text.isdigit() else text
30
31 def natural_keys(text):
32     return [ atoi(c) for c in re.split(r'(\d+)', text) ]
33
34
35 # In[3]:
36
37
38 def my_index(l, x, default=False):
39     if x in l:
40         return l.index(x)
41     else:
42         return default
43
44
45 # In[4]:
46
47
48 def TimeLabel(f1):
49     i_1=f1.index('Mesh_Block_0001')
50     if not my_index(f1[i_1+1:], 'Mesh_Block_0001'):
51         times=np.array([float(f1[i_1+1].split()[2].strip('t='))])
52         TimeBlock=[ f1[i_1:]]
53     else:
54         di=f1[i_1+1:].index('Mesh_Block_0001')+1
55         TBS=np.arange(i_1, len(f1), di)
56         times=np.array([float(f1[i+1].split()[2].strip('t=')) for i in TBS ])
57         TimeBlock=[ f1[i:i+di] for i in TBS]
58     return times, TimeBlock
59
60
61 # In[5]:
62
63
64 def MBdata (MeshBlocks):
65     data=[dict() for MB in MeshBlocks]
66     for i, MB in enumerate(MeshBlocks):
67         a=np.array([values.split() for values in MB]).astype('float64')
68         x=np.unique(a[:,0])
69         y=np.unique(a[:,1])
70         z=np.unique(a[:,2])
71         variable=a[:,3].reshape(z.size,y.size,x.size)
72         data[i]['x']=x
73         data[i]['y']=y
74         data[i]['z']=z
75         data[i]['variable']=variable
```





```
174     @classmethod
175     def get_TimeSeries(self,i_x,i_y):
176         return TimeSeries([self.data[i_t][0][i_y,i_x] for i_t in range(self.nt)],
177                             t0=self.t0,dt=self.dt,unit=self.unit,
178                             name='x = {:} m, y = {:} m'.format(self.MBs[0].x[i_x],
179                             self.MBs[0].y[i_y]))
179     '''
180     #があると使いませない@classmethod
181     def read_txt(self,filename,variable):
182         f1=open(filename).read().splitlines()
183         f1.append('')
184         times,TimeBlock=TimeLabel(f1)
185         MBs=MakeMB(TimeBlock[0],variable)
186         data=[]
187         for TB in tqdm(TimeBlock):
188             data.append(Make_data(TB,variable))
189         unit=''
190         self.variable=copy.deepcopy(variable)
191         self.data=copy.deepcopy(data)
192         self.times=copy.deepcopy(times)
193         self.MBs=copy.deepcopy(MBs)
194         self.unit=copy.deepcopy(unit)
195         self.t0=copy.deepcopy(times[0])
196         self.nt=copy.deepcopy(times.size)
197         if times.size==1:
198             self.dt=copy.deepcopy(0)
199         else:
200             self.dt=copy.deepcopy(1/round(1/((times[-1]-times[0])/times.size)))
201         return self
202
203
204     # In[10]:
205
206
207     #時系列データの取得
208     def get_TimeSeries(flowdata,i_x,i_y,i_z):
209         return TimeSeries([flowdata.data[i_t][0][i_z,i_y,i_x] for i_t in range(
210             flowdata.nt)],
211                             t0=flowdata.t0,dt=flowdata.dt,unit=flowdata.unit,
212                             name='x_u={:}um, y_u={:}um, z_u={:}um'.format(flowdata.MBs
213                                 [0].x[i_x], flowdata.MBs[0].y[i_y], flowdata.MBs[0].z[i_z]
214                                 ))
215
216
217     # In[11]:
218
219     """
220     def get_truedata_test(flowdata,default):
221         [[ default in flowdata.data[i_t][0][:,i_y,i_x]
```

```

220         for i_x in range(flowdata.MBs[0].nx)] for i_y in range(flowdata.MBs[0].ny)]
221         for i_t in range(flowdata.nt)]
222     """
223
224
225     # In[12]:
226
227
228     #ある地点、ある時間のフルード数を計算
229     def pointFroude(flowdata, i_y=10, i_z=1, i_x=500, i_t=100):
230         g=9.8
231         celllength=0.02
232         velocity=flowdata.data[i_t][0][i_z, i_y, i_x]
233         #depth=celllength*(np.count_nonzero(flowdata.data[i_t][0][:, i_y, i_x], axis=0)-
234             i_z)
235         depth=celllength*(i_z-0.5)
236         Froude=velocity/(depth*g)**0.5
237         return Froude
238
239     # In[13]:
240
241
242     #ある地点のフルード数の時間経過、方向にポイントx
243     def pointFroudeTimeseriesX(flowdata, i_y=10, i_z=1, i_xmin=0, i_xmax=100, n_x=20,
244         t_start=200, t_end=300):
245         g=9.8
246         celllength=0.015
247         R=0.2
248         TS=[]
249         TS2=[]
250         fig = Plot(figsize=(10, 5))
251         ax = fig.gca()
252         di_x=int((i_xmax+1-i_xmin)/n_x)
253         y=abs(R-(celllength*(i_y+0.5)))
254         depth=celllength*(i_z+0.5)-(R-(R**2-y**2)**0.5)
255         for i_x in range(i_xmin, i_xmax+1, di_x):
256             TS.append(get_TimeSeries(flowdata, i_x, i_y, i_z))
257             TS2.append(TS[-1]/(depth*g)**0.5)
258             color = cm.viridis(float((i_x-i_xmin)/(i_xmax-i_xmin)))
259             ax.plot(TS2[-1], color=color, label='xU={:}μm'.format( flowdata.MBs[0].x[
260                 i_x]))
261
262         ax.set_title('yU={:}μm, zU={:}μm'.format( flowdata.MBs[0].y[i_y], flowdata.MBs
263             [0].z[i_z]))
264         ax.set_xscale('auto-gps', epoch=0)
265         ax.set_ylabel(' [m/s ] ')
266         ax.legend(bbox_to_anchor=(1.05, 1), loc='upperUleft' )
267         ax.set_xlim(t_start, t_end)
268         return fig, TS2

```

```
266
267
268 # In[14]:
269
270
271 #ある地点のフルード数の時間経過、方向にポイントy
272 def pointFroudeTimeseriesY(flowdata,i_x=500,i_z=1,i_ymin=0,i_ymax=19,n_y=10,
    t_start=200,t_end=300):
273     g=9.8
274     celllength=0.015
275     R=0.2
276     TS2=[]
277     TS22=[]
278     fig = Plot(figsize=(10, 5))
279     ax = fig.gca()
280     di_y=int((i_ymax+1-i_ymin)/n_y)
281     for i_y in range(i_ymin,i_ymax+1,di_y):
282         y=abs(R-(celllength*(i_y+0.5)))
283         depth=celllength*(i_z+0.5)-(R-(R**2-y**2)**0.5)
284         TS2.append(get_TimeSeries(flowdata,i_x,i_y,i_z))
285         TS22.append(TS2[-1]/(depth*g)**0.5)
286         color = cm.viridis(float((i_y-i_ymin)/(i_ymax-i_ymin)))
287         ax.plot(TS22[-1],color=color,label='yU={:}UM'.format(flowdata.MBs[0].y[
            i_y]))
288
289     ax.set_title('xU={:}UM,zU={:}UM'.format(flowdata.MBs[0].x[i_x],flowdata.MBs
        [0].z[i_z]))
290     ax.set_xscale('auto-gps',epoch=0)
291     ax.set_ylabel('[m/s]')
292     ax.legend(bbox_to_anchor=(1.05, 1), loc='upper_left' )
293     ax.set_xlim(t_start,t_end)
294     return fig,TS22
295
296
297 # In[15]:
298
299
300 #ある地点のフルード数の時間経過、方向にポイントz
301 def pointFroudeTimeseriesZ(flowdata,i_x=500,i_y=10,i_zmin=0,i_zmax=10,n_z=10,
    t_start=200,t_end=300):
302     g=9.8
303     celllength=0.015
304     R=0.2
305     TS3=[]
306     TS32=[]
307     fig = Plot(figsize=(10, 5))
308     ax = fig.gca()
309     di_z=int((i_zmax+1-i_zmin)/n_z)
310     y=abs(R-(celllength*(i_y+0.5)))
311     for i_z in range(i_zmin,i_zmax+1,di_z):
```

```
312         depth=celllength*(i_z+0.5)-(R-(R**2-y**2)**0.5)
313         TS3.append(get_TimeSeries(flowdata,i_x,i_y,i_z))
314         TS32.append(TS3[-1]/(depth*g)**0.5)
315         color = cm.viridis(float((i_z-i_zmin)/(i_zmax-i_zmin)))
316         ax.plot(TS32[-1],color=color,label='z_{}_m'.format(flowdata.MBs[0].z[
317             i_z]))
318
319     ax.set_title('x_{}_m,y_{}_m'.format(flowdata.MBs[0].x[i_x],flowdata.MBs
320         [0].y[i_y]))
321     ax.set_xscale('auto-gps',epoch=0)
322     ax.set_ylabel('[m/s]')
323     ax.legend(bbox_to_anchor=(1.05, 1), loc='upper_left' )
324     ax.set_xlim(t_start,t_end)
325     return fig,TS32
326
327 # In[16]:
328
329 #ある地点の時系列データ、方向にポイントx
330 def PlotTimeseries_ix(flowdata,i_y=10,i_z=1,i_xmin=0,i_xmax=100,n_x=20,t_start
331     =200,t_end=300):
332     TS=[]
333     fig = Plot(figsize=(10, 5))
334     ax = fig.gca()
335     di_x=int((i_xmax+1-i_xmin)/n_x)
336     for i_x in range(i_xmin,i_xmax+1,di_x):
337         TS.append(get_TimeSeries(flowdata,i_x,i_y,i_z))
338         color = cm.viridis(float((i_x-i_xmin)/(i_xmax-i_xmin)))
339         ax.plot(TS[-1],color=color,label='x_{}_m'.format(flowdata.MBs[0].x[i_x
340             ]))
341
342     ax.set_title('y_{}_m,z_{}_m'.format(flowdata.MBs[0].y[i_y],flowdata.MBs
343         [0].z[i_z]))
344     ax.set_xscale('auto-gps',epoch=0)
345     ax.set_ylabel('[m/s]')
346     ax.legend(bbox_to_anchor=(1.05, 1), loc='upper_left' )
347     ax.set_xlim(t_start,t_end)
348     return fig,TS
349
350 # In[17]:
351
352 #ある地点の時系列データ、方向にポイントy
353 def PlotTimeseries_iy(flowdata,i_x=500,i_z=1,i_ymin=0,i_ymax=19,n_y=10,t_start
354     =200,t_end=300):
355     TS2=[]
356     fig = Plot(figsize=(10, 5))
357     ax = fig.gca()
```

```
356     di_y=int((i_ymax+1-i_ymin)/n_y)
357     for i_y in range(i_ymin,i_ymax+1,di_y):
358         TS2.append(get_TimeSeries(flowdata,i_x,i_y,i_z))
359         color = cm.viridis(float((i_y-i_ymin)/(i_ymax-i_ymin)))
360         ax.plot(TS2[-1],color=color,label='yU=U{:}μm'.format( flowdata.MBs[0].y[
361             i_y]))
362
363     ax.set_title('xU=U{:}μm,zU=U{:}μm'.format( flowdata.MBs[0].x[i_x],flowdata.MBs
364         [0].z[i_z]))
365     ax.set_xscale('auto-gps',epoch=0)
366     ax.set_ylabel('[m/s]')
367     ax.legend(bbox_to_anchor=(1.05, 1), loc='upper_left' )
368     ax.set_xlim(t_start,t_end)
369     return fig,TS2
370
371 # In[18]:
372
373 #ある地点の時系列データ、方向にポイントz
374 def PlotTimeseries_iz(flowdata,i_x=500,i_y=10,i_zmin=0,i_zmax=10,n_z=10,t_start
375     =200,t_end=300):
376     TS3=[]
377     fig = Plot(figsize=(10, 5))
378     ax = fig.gca()
379     di_z=int((i_zmax+1-i_zmin)/n_z)
380     for i_z in range(i_zmin,i_zmax+1,di_z):
381         TS3.append(get_TimeSeries(flowdata,i_x,i_y,i_z))
382         color = cm.viridis(float((i_z-i_zmin)/(i_zmax-i_zmin)))
383         ax.plot(TS3[-1],color=color,label='zU=U{:}μm'.format( flowdata.MBs[0].z[
384             i_z]))
385
386     ax.set_title('xU=U{:}μm,yU=U{:}μm'.format( flowdata.MBs[0].x[i_x],flowdata.MBs
387         [0].y[i_y]))
388     ax.set_xscale('auto-gps',epoch=0)
389     ax.set_ylabel('[m/s]')
390     ax.legend(bbox_to_anchor=(1.05, 1), loc='upper_left' )
391     ax.set_xlim(t_start,t_end)
392     return fig,TS3
393
394 # In[19]:
395
396 #パイプ断面(平面yz)のパラメータの平均は抜かす(0)
397 def average_velocity_yz(flowdata,i_t=200):
398     data_ave=[]
399     fig = Plot(figsize=(15, 5))
400     ax = fig.gca()
401     ax.set_xlabel("x [m] ")
```

```

401     ax.set_ylabel("velocity[m/s]")
402     ax.set_title('t_□={:}□s'.format(flowdata.times[i_t]))
403     for i_mb in range(len(flowdata.MBs)):
404         nandata=np.where(flowdata.data[i_t][i_mb]==0,np.nan,flowdata.data[i_t][
            i_mb]) #を0に変
            換nan
405         data_ave.append(np.nanmean(nandata,axis=(0,1))) #を抜かして平均をとるnan
406         ax.plot(flowdata.MBs[i_mb].x,data_ave[-1])
407     return fig,data_ave
408
409
410 # In[20]:
411
412
413 #パイプ断面(平面yz)の平均水深の計算、パラメータは流体の存在部分にあるならなんでもいい
414 def average_hyfd(flowdata,i_t=200,i_y=10):
415     celllength=0.02
416     R=0.2
417     hyfd_ave=[]
418     fig = Plot(figsize=(15, 5))
419     ax = fig.gca()
420     ax.set_xlabel("x[m]")
421     ax.set_ylabel("depth(average) [m]")
422     ax.set_title('t_□={:}□s'.format(flowdata.times[i_t]))
423     for i_mb in range(len(flowdata.MBs)):
424         nonzeromesh=np.count_nonzero(flowdata.data[i_t][i_mb],axis=(0,1))
425         water_horizon=2*((R**2-np.abs(R-celllength*np.count_nonzero(flowdata.data[
            i_t][i_mb][:,i_y,:],axis=0))**2)**0.5) #一番上の横
            幅
426         average_hyfd=nonzeromesh*(celllength**2)/water_horizon
427         hyfd_ave.append(average_hyfd)
428         ax.plot(flowdata.MBs[i_mb].x,hyfd_ave[-1])
429     return fig,hyfd_ave
430
431
432 # In[21]:
433
434
435 #ある時間の平均水深と平均流速からのフルード数、方向にポイントx
436 def Frude_xdirection(flowdata,i_t=199):
437     velocity_ave=average_velocity_yz(flowdata,i_t)[1]
438     hyfd_ave=average_hyfd(flowdata,i_t)[1]
439     g=9.8 #重力加速度
440     Frude=[]
441     fig = Plot(figsize=(15, 5))
442     ax = fig.gca()
443     ax.set_xlabel("x[m]")
444     ax.set_ylabel("Frude_□number [m]")
445     ax.set_title('t_□={:}□s'.format(flowdata.times[i_t]))
446     for i_mb in range(len(flowdata.MBs)):
447         Frude.append(velocity_ave[i_mb]/(g*hyfd_ave[i_mb])**0.5)

```

```
448         ax.plot(flowdata.MBs[i_mb].x,Frude[-1])
449         ax.hlines(1,flowdata.MBs[i_mb].x[0],flowdata.MBs[i_mb].x[-1],color="
           red")
450     return fig,Frude
451
452
453 # In [22]:
454
455
456 #パイプ断面の平均速度、の縦軸時間、横軸方向のスペクトログラムx
457 def PlotSpectrogram_x(flowdata,t_start=290,t_end=300,datamin=0.8,datamax=1.2):
458     data=np.vstack([list(itertools.chain.from_iterable(average_velocity_yz(
           flowdata,i_t=t)[1])) for t in range(flowdata.nt)])
459     #同じ時刻のデータは横一列次元(1)に直した
460     fig = Plot(figsize=(15, 5))
461     ax = fig.gca()
462     ax.set_xlabel("x[m]")
463     ax.set_ylabel("t[s]")
464     #ax.set_xlim(0,13)
465     ax.set_title('t={:}-{:}s'.format(t_start,t_end))
466     ax.imshow(data,
467               extent=[flowdata.MBs[0].x[0],
468                       flowdata.MBs[-1].x[-1],
469                       flowdata.times[0],
470                       flowdata.times[-1]],
471               vmin=datamin, vmax=datamax, aspect='auto', origin='lower',
472               interpolation='nearest')
473
474     return fig,data
475
476
477 # In [23]:
478
479 #平均フルード数、の縦軸時間、横軸方向のスペクトログラム、ただし以上以下の色分けx1
480 def PlotSpectrogram_x_Frude(flowdata,t_start=290,t_end=300,datamin=0,datamax=2):
481     data=np.vstack([list(itertools.chain.from_iterable(Frude_xdirection(flowdata,
           i_t=t)[1])) for t in range(flowdata.nt)])
482     #同じ時刻のデータは横一列次元(1)に直した
483     fig = Plot(figsize=(15, 5))
484     ax = fig.gca()
485     ax.set_xlabel("x[m]")
486     ax.set_ylabel("t[s]")
487     #ax.set_xlim(0,13)
488     ax.set_title('t={:}-{:}s'.format(t_start,t_end))
489     cmap = colors.ListedColormap(['white', 'red'])
490     bounds=[0,1,10]
491     norm = colors.BoundaryNorm(bounds, cmap.N)
492     ax.imshow(data,
493               extent=[flowdata.MBs[0].x[0],
494                       flowdata.MBs[-1].x[-1],
```

```
494         flowdata.times[0],
495         flowdata.times[-1]],
496         vmin=datamin, vmax=datamax, aspect='auto', origin='lower',
           interpolation='nearest', cmap=cmap, norm=norm)
497     return fig, data
498 #常流射流判別用
499
500
501 # In[24]:
502
503
504 #平均フルード数、の縦軸時間、横軸方向のスペクトログラムx
505 def PlotSpectrogram_x_Frude_value(flowdata, t_start=290, t_end=300, datamin=0, datamax
    =2):
506     data=np.vstack([list(itertools.chain.from_iterable(Frude_xdirection(flowdata,
        i_t=t)[1])) for t in range(flowdata.nt)])
507     #同じ時刻のデータは横一列次元(1)に直した
508     fig = Plot(figsize=(15, 5))
509     ax = fig.gca()
510     ax.set_xlabel("x[m]")
511     ax.set_ylabel("t[s]")
512     #ax.set_xlim(0,13)
513     ax.set_title('t={:}-{:}s'.format(t_start, t_end))
514     ax.imshow(data,
515               extent=[flowdata.MBs[0].x[0],
516                       flowdata.MBs[-1].x[-1],
517                       flowdata.times[0],
518                       flowdata.times[-1]],
519               vmin=datamin, vmax=datamax, aspect='auto', origin='lower',
           interpolation='nearest')
520     return fig, data
521
522
523 # In[25]:
524
525
526 #以下2データ限定D(を通すと使えるget_2Ddata)かつつのテキストデータを読み込んだ場合(1)
527 #複数のテキストデータを読み込んだ場合はを通すと使える。WaterTimeseries_point_all
528
529
530 # In[26]:
531
532
533 def get_TimeSeries_2D(flowdata, i_x, i_y):
534     return TimeSeries([flowdata.data[i_t][0][i_y, i_x] for i_t in range(flowdata.nt
        )]),
535                       t0=flowdata.t0, dt=flowdata.dt, unit=flowdata.unit,
536                       name='xu=u{:}um, yu=u{:}um'.format(flowdata.MBs[0].x[i_x],
           flowdata.MBs[0].y[i_y]))
537
```

```
538
539 # In[27]:
540
541
542 #次元データを次元プロット複数のメッシュに使える。22
543 def plot_2D(flowdata,t_i,min,max):
544     fig = Plot(figsize=(15, 5))
545     ax = fig.gca()
546     ax.set_xlabel("x[m]")
547     ax.set_ylabel("y[m]")
548     ax.set_title('t0={}s'.format(flowdata.times[t_i]))
549     for data,mb in zip(flowdata.data[t_i],flowdata.MBs):
550         ax.imshow(data, extent=mb.extent, aspect='auto', interpolation='nearest',
551                 clim=[min,max])
552     fig.colorbar(label=r'{}''.format(''))
553     return fig
554
555 # In[28]:
556
557
558 #ある地点の時系列データ、方向にポイント、限定x2D
559 def PlotTimeseries_ix_2D(flowdata,i_y=13,i_xmin=0,i_xmax=800,n_x=20,t_start=900,
560     t_end=1000):
561     TS=[]
562     fig = Plot(figsize=(10, 5))
563     ax = fig.gca()
564     di_x=int((i_xmax+1-i_xmin)/n_x)
565     for i_x in range(i_xmin,i_xmax+1,di_x):
566         TS.append(get_TimeSeries_2D(flowdata,i_x,i_y))
567         color = cm.viridis(float((i_x-i_xmin)/(i_xmax-i_xmin)))
568         ax.plot(TS[-1],color=color,label='x0={}m,y0={}m'.format(flowdata.
569             MBs[0].x[i_x],flowdata.MBs[0].y[i_y]))
570
571     ax.set_title('Water0level0Time0series')
572     ax.set_xscale('auto-gps',epoch=0)
573     ax.set_ylabel('[m]')
574     ax.legend(bbox_to_anchor=(1.05, 1), loc='upper0left')
575     ax.set_xlim(t_start,t_end)
576     return fig,TS
577
578 # In[29]:
579
580 #ある地点の時系列データ、方向にポイント、限定y2D
581 def PlotTimeseries_iy_2D(flowdata,i_x=800,i_ymin=0,i_ymax=19,n_y=10,t_start=900,
582     t_end=1000):
583     TS2=[]
584     fig = Plot(figsize=(10, 5))
```

```

584     ax = fig.gca()
585     di_y=int((i_ymax+1-i_ymin)/n_y)
586     for i_y in range(i_ymin,i_ymax+1,di_y):
587         TS2.append(get_TimeSeries_2D(flowdata,i_x,i_y))
588         color = cm.viridis(float((i_y-i_ymin)/(i_ymax-i_ymin)))
589         ax.plot(TS2[-1],color=color,label='y={:}μm,x={:}μm'.format( flowdata.
                    MBs[0].y[i_y],flowdata.MBs[0].x[i_x]))
590
591     ax.set_title('Water_level_Time_series')
592     ax.set_xscale('auto-gps',epoch=0)
593     ax.set_ylabel('[m]')
594     ax.legend(bbox_to_anchor=(1.05, 1), loc='upper_left' )
595     ax.set_xlim(t_start,t_end)
596     return fig,TS2
597
598
599 # In[30]:
600
601
602 #次元データの各地点の時系列データをフーリエ変換して、各地点のスペクトルを計算2
603 #上の関数で出せる時系列データを利用する。
604 def PlotASDs(TS_list,fftlength=10,t_start=900,t_end=1000):
605     fig = Plot(figsize=(10, 5))
606     ax = fig.gca()
607     n_TS=len(TS_list)
608     ASD=[]
609     for i,temp in enumerate(TS_list):
610         ASD.append(temp.crop(t_start,t_end).asd(fftlength,fftlength/2,method='
                    Welch'))
611         color = cm.viridis(float(i/n_TS))
612         ax.plot(ASD[-1],color=color,label=temp.name)
613
614     ax.set_title('t={:}-{:}s'.format(t_start,t_end))
615     ax.set_xlabel('frequency [Hz]')
616     ax.set_ylabel('Water_level_ASD [m/rtHz]')
617     ax.legend(bbox_to_anchor=(1.05, 1), loc='upper_left' )
618     ax.set_xscale('log')
619     ax.set_yscale('log')
620     return fig,ASD
621
622
623 # In[31]:
624
625
626 #次元データの各地点の時系列データ2(TS)を比べてコヒーレンスを、スペクトログラムではないplot
627 def PlotCoherence(TS_list,i_0,fftlength=10,t_start=900,t_end=1000):
628     fig = Plot(figsize=(10, 5))
629     ax = fig.gca()
630     n_TS=len(TS_list)
631     for i,temp in enumerate(TS_list):

```

```
632         color = cm.viridis(float(i/n_TS))
633         ax.plot(temp.crop(t_start,t_end).coherence(TS[i_0].crop(t_start,t_end),
634                                                     fftlength,fftlength/2),color=
                                                     color,label=temp.name)
635
636     ax.legend()
637     ax.set_xlabel('frequency [Hz]')
638     ax.set_ylabel('coherence^2')
639     ax.set_xscale('log')
640     ax.set_title(TS[i_0].name+',_t={:}-{:}s'.format(t_start,t_end))
641     ax.legend(bbox_to_anchor=(1.05, 1), loc='upper_left' )
642     return fig
643
644
645 # In[32]:
646
647
648 #次元データの各地点の2をスペクトログラム表示、ただし表示範囲に注意そのまま使えないことが多い。その場合はアレ
649 #ンジをASD
650 def PlotSpectrogram_x(flowdata,i_y=10,t_start=900,t_end=100,vmin=1e-9,vmax=1e-4,
651                       fftlength=10):
652     asd=[get_TimeSeries_2D(flowdata,i_x,i_y).crop(t_start,t_end).asd(fftlength,
653                               fftlength/2,method='Welch') for i_x in range(flowdata.MBs[0].nx)]
654     ASD_xf=np.vstack([a.value for a in asd]).T
655     fig = Plot(figsize=(15, 5))
656     ax = fig.gca()
657     ax.set_xlabel("x[m]")
658     ax.set_ylabel("f [Hz]")
659     ax.set_yscale('log')
660     ax.set_ylim(0.1,10)
661     ax.set_title('ASD_y={:}_t={:}-{:}s'.format( flowdata.MBs[0].y[i_y],
662                                               t_start,t_end))
663     im=ax.imshow(ASD_xf,
664                 extent=[flowdata.MBs[0].x[0],
665                         flowdata.MBs[0].x[-1],
666                         asd[0].frequencies[0].value,
667                         asd[0].frequencies[-1].value],
668                 norm='log',vmin=vmin,vmax=vmax,aspect='auto',origin='lower',
669                 interpolation='nearest')
670     return fig.colorbar(im,ax=ax)
671
672 # In[33]:
673
674 #ある地点と他の地点のコヒーレンスをスペクトログラムで表示
675 def PlotSpectrogram_Coherence(flowdata,i_y=10,i_x_0=400,t_start=900,t_end=1000,
676                               fftlength=10):
677     TS0=get_TimeSeries_2D(flowdata,i_x_0,i_y).crop(t_start,t_end)
678     coh=[get_TimeSeries_2D(flowdata,i_x,i_y).crop(t_start,t_end).coherence(TS0,
```

```

        fftlength,fftlength/2) for i_x in range(flowdata.MBs[0].nx)]
675 COH_xf=np.vstack([a.value for a in coh]).T
676 fig = Plot(figsize=(15, 5))
677 ax = fig.gca()
678 ax.set_xlabel("x[m]")
679 ax.set_ylabel("f[Hz]")
680 ax.set_yscale("log")
681 ax.set_ylim(0.1,10)
682 ax.set_title('Coherence-x={:}y={:}z={:}m'.format( flowdata.MBs[0].x[i_x_0],
        flowdata.MBs[0].y[i_y]))
683 im=ax.imshow(COH_xf ,
684               extent=[flowdata.MBs[0].x[0] ,
685                       flowdata.MBs[0].x[-1] ,
686                       coh[0].frequencies[0].value ,
687                       coh[0].frequencies[-1].value] ,
688               aspect='auto', origin='lower', interpolation='nearest')
689 ax.axvline(flowdata.MBs[0].x[i_x_0],color='red')
690 return fig.colorbar(im, ax=ax)
691
692
693 # In[33]:
694
695
696 #以下3変換と座標計算とニュートニアン計算D
697
698
699 # In[34]:
700
701
702 #3データをデータに変換、方向の配列はないD2Dz
703 #元々2として読み込んだデータも方向の配列が残っているのでここを通して消す必要がある。Dz1bin
704 def get_2Ddata(flowdata,default):
705     New_timeseries_data=[]
706     for i_t in range(flowdata.nt):
707         New_MB_data=[]
708         for i_MB in range(len(flowdata.MBs)):
709             New_y_data=[]
710             for i_y in range(flowdata.MBs[i_MB].ny):
711                 New_x_data=[]
712                 for i_x in range(flowdata.MBs[i_MB].nx):
713                     for i_z in range(flowdata.MBs[i_MB].nz):
714                         truedata= flowdata.data[i_t][i_MB][i_z,i_y,i_x]
715                         if truedata!=default:
716                             break
717                         New_x_data.append(truedata)
718                         New_y_data.append(New_x_data)
719                     New_MB_data.append(np.array(New_y_data).astype('float64'))
720                 New_timeseries_data.append(New_MB_data)
721 data=New_timeseries_data
722 times=flowdata.times

```

```
723     MBs=flowdata.MBs
724     t0=flowdata.times[0]
725     nt=flowdata.times.size
726     unit=flowdata.unit
727     if times.size==1:
728         dt=0
729     else:
730         dt=1/round(1/((times[-1]-times[0])/flowdata.times.size))
731     return FlowData(data,times,MBs,unit)
732
733
734 # In[35]:
735
736
737 #水の重心座標計算、水面座標をついでに導出
738 def hyfd_to_Halflevel(hyfd,terrainelevation):
739     data=copy.deepcopy(hyfd.data)
740     times=copy.deepcopy(hyfd.times)
741     MBs=copy.deepcopy(hyfd.MBs)
742     unit=copy.deepcopy(hyfd.unit)
743     data2=copy.deepcopy(terrainelevation.data)
744     times2=copy.deepcopy(terrainelevation.times)
745     MBs2=copy.deepcopy(terrainelevation.MBs)
746     unit2=copy.deepcopy(terrainelevation.unit)
747
748     LevelData=[[data[i_t][i_MB]+data2[0][i_MB] for i_MB in range(len(MBs))] for
749                 i_t in range(times.size)]
750     HalfLevelData=[[data[i_t][i_MB]/2+data2[0][i_MB] for i_MB in range(len(MBs))]
751                    for i_t in range(times.size)]
752     return FlowData(HalfLevelData,times,MBs,unit),FlowData(LevelData,times,MBs,
753                    unit)
754
755 # In[36]:
756
757 #ニュートニアン計算
758 #メッシュブロックの時にしか使えないことに注意1
759 def Newtonian(hyfd,HalfLevelData,ix_min=0,ix_max=0,iy_min=0,iy_max=0,t_start=900,
760               t_end=1000):
761     G=6.67408*10**(-11) #万有引力定数
762     rho=1000
763     x_TM=0
764     y_TM=5
765     z_TM=1.5 #[m]
766     R=0.2 #[m]
767     data=copy.deepcopy(hyfd.data)
768     times=copy.deepcopy(hyfd.times)
769     MBs=copy.deepcopy(hyfd.MBs)
770     unit=copy.deepcopy(hyfd.unit)
```

```
769     data2=copy.deepcopy(HalfLevelData.data)
770     times2=copy.deepcopy(HalfLevelData.times)
771     MBs2=copy.deepcopy(HalfLevelData.MBs)
772     unit2=copy.deepcopy(HalfLevelData.unit)
773     ACC=np.zeros(times.size)
774     i_MB=0
775     #     for i_MB in range(len(MBs)):
776     if ix_max==0:
777         ix_max=MBs[i_MB].nx-1
778     if iy_max==0:
779         iy_max=MBs[i_MB].ny-1
780     for i_t in range(times.size):
781         r3=((x_TM-MBs[i_MB].xx)**2+(y_TM-MBs[i_MB].yy)**2+(z_TM-data2[i_t][i_MB])
782             **2)**1.5
783         dACC=G*rho*data[i_t][i_MB]*(MBs[i_MB].xx-x_TM)/r3
784         ACC[i_t] = integrate.trapz(integrate.trapz(dACC[iy_min:iy_max+1,ix_min:
785             ix_max+1], MBs[i_MB].y[iy_min:iy_max+1],axis=0),MBs[i_MB].x[ix_min:
786             ix_max+1])
787         ACC_time=TimeSeries(ACC,t0=hyfd.t0,dt=hyfd.dt,unit='m/s^2')
788         ACC_ASD=ACC_time.crop(t_start,t_end).asd(10,5)
789         Strain_ASD=ACC_ASD/(2*pi*ACC_ASD.frequencies)**2/3000
790         return ACC_time,ACC_ASD,Strain_ASD
791
792
793     # In[37]:
794
795     terrainelevation=FlowData()
796
797     # In[38]:
798
799     terrainelevation.read_txt("geodata_20mpipe_FR36_2_Tilt03_K32mm_Mesh0015_20Hz.txt",
800                             "terele")
801
802     # In[39]:
803
804     terrainelevation2D=get_2Ddata(terrainelevation,-0.0013230224)
805
806     # In[40]:
807
808     files = sorted(glob.glob("20mpipe_FR36_2_Tilt03_K32mm_Mesh0015_20Hz_*.txt"), key=
809                   natural_keys)
810     print(files)
811     # ファイルの番号が若い順から記録されるのに注意、後ろのコードでは若い番号を先頭に時系列データができる
```

```
814
815
816 # In[41]:
817
818
819 hyfd2D=[]
820 Halfleveldata=[]
821 water_level=[]
822 ACC_time=[]
823 ACC_ASD=[]
824 Strain_ASD=[]
825 for file in files:
826     hyfd=FlowData() #前の結果を上書き
827     hyfd.read_txt(file,"hyfd")
828     hyfd2D.append(get_2Ddata(hyfd,0))
829     Halfleveldata.append(hyfd_to_Halflevel(hyfd2D[-1],terrinelevation2D)[0])
830     water_level.append(hyfd_to_Halflevel(hyfd2D[-1],terrinelevation2D)[1])
831     t_s=Halfleveldata[-1].times[0]
832     t_e=Halfleveldata[-1].times[-1]
833     ACC_time.append(Newtonian(hyfd2D[-1],Halfleveldata[-1],t_start=t_s,t_end=t_e)
834                        [0])
835     ACC_ASD.append(Newtonian(hyfd2D[-1],Halfleveldata[-1],t_start=t_s,t_end=t_e)
836                      [1])
837     Strain_ASD.append(Newtonian(hyfd2D[-1],Halfleveldata[-1],t_start=t_s,t_end=t_e)
838                        [2])
839 with open("20mpipe_FR36_2_Tilt03_K32_3D_20Hz_hyfd2.pkl","wb") as f:
840     pickle.dump(hyfd2D, f)
841 with open("20mpipe_FR36_2_Tilt03_K32_3D_20Hz_Halfleveldata.pkl","wb") as f:
842     pickle.dump(Halfleveldata, f)
843 with open("20mpipe_FR36_2_Tilt03_K32_3D_20Hz_water_level.pkl","wb") as f:
844     pickle.dump(water_level, f)
845
846
847 # In[42]:
848
849 #再計算用、を読み直して実行NewtonianNewtonian
850 def Newtonian_2(hyfd2D,Halfleveldata,i_t_start=0,i_t_end=20):
851     ACC_time=[]
852     ACC_ASD=[]
853     Strain_ASD=[]
854     for t in range(i_t_start,i_t_end):
855         t_s=Halfleveldata[t].times[0]
856         t_e=Halfleveldata[t].times[-1]
857         ACC_time.append(Newtonian(hyfd2D[t],Halfleveldata[t],t_start=t_s,t_end=t_e)
858                          [0])
859         ACC_ASD.append(Newtonian(hyfd2D[t],Halfleveldata[t],t_start=t_s,t_end=t_e)
860                          [1])
861         Strain_ASD.append(Newtonian(hyfd2D[t],Halfleveldata[t],t_start=t_s,t_end=
862                                 t_e)[2])
```

```
858     return ACC_time, ACC_ASD, Strain_ASD
859
860
861 # In[43]:
862
863
864 ACC_time_all=np.concatenate(ACC_time) #すべてを合計
865 with open("20mpipe_FR36_2_Tilt03_K32_20Hz_Newtonian.pkl","wb") as f:
866     pickle.dump(ACC_time_all, f)
867
868
869 # In[44]:
870
871
872 ACC_time_all
873
874
875 # In[45]:
876
877
878 ACC_ASD_all=ACC_time_all.crop(900,1000).asd(10,5)
879
880
881 # In[46]:
882
883
884 Strain_ASD_all=ACC_ASD_all/(2*pi*ACC_ASD_all.frequencies)**2/3000
885
886
887 # In[47]:
888
889
890 '''
891 z0=hyfd2D.data[-1][0][:,300]
892 y1=hyfd2D.MBs[0].y
893 z5=terrainelevation2D.data[0][0][:,300]
894 z6=z0+z5
895 z7=Halfleveldata.data[-1][0][:,300]
896 '''
897
898
899 # In[48]:
900
901
902 '''
903 #plt.plot(y1, z0)
904 #plt.plot(y1, z1,label='offset(hands)',color='black')
905 #plt.plot(y1, z2,label='Water level(offset(hand)+flow depth)',color='blue')
906 #plt.plot(y1, z3,label='Half water level.old',color='orange')
907 #plt.plot(y1, z4,label='Free surface elevation',color='red')
```

```
908 plt.plot(y1, z5, label='offset (FLOW-3D)', color='purple')
909 plt.plot(y1, z6, label='Water level(offset (FLOW-3D)+flow depth)', color='green')
910 plt.plot(y1, z7, label='Half water level', color='blue')
911 #plt.ylim(-0.11, -0.095)
912 plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
913 '''
914
915
916 # In[49]:
917
918
919 ACC_time_all.plot(xscale='auto-gps', epoch=0, xlim=(900, 1000))
920
921
922 # In[50]:
923
924
925 ACC_time_all.times
926
927
928 # In[51]:
929
930
931 ACC_ASD_all.plot(title='Acceleration ASD')
932
933
934 # In[52]:
935
936
937 Strain_ASD_all.plot(ylabel='Strain [/rtHz]', title='Newtonian Noise')
938
939
940 # In[62]:
941
942
943 KAGRA=FrequencySeries.read('DKAGRA.txt')
944 ET_sum=FrequencySeries.read('ET-0000A-18_ETDSensitivityCurveTxtFile_sum.txt')
945 plot = Plot(figsize=(6, 4))
946 ax = plot.add_subplot(xscale='log', yscale='log')
947 ax.plot(Strain_ASD_all, label='36.2t/h')
948 ax.plot(KAGRA, label='KAGRA design sensitivity')
949 #ax.plot(ET_sum, label='ET design sensitivity')
950 ax.set_ylabel('Strain [/rtHz]')
951 ax.set_xlabel('Frequency [Hz]')
952 ax.set_title('Newtonian Noise (straight pipe)')
953 ax.set_xlim(0.1, 10)
954 ax.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
955
956
957 # In[50]:
```

```
958
959
960 #複数のテキストデータを読み込んだ場合
961 #かつされたものに対して使える。get_2Ddataこれは上記コードではされてるはずである()
962
963
964 # In [51]:
965
966
967 #複数テキストデータを読み込んだ場合のリストの指定したある一地点をテキストデータごとリストではなくサンプリング時間ごとのリストにする。'''
968 def WaterTimeseries_point(water_level,i_x=0,i_y=0):
969     #ある座標の水面変化の時系列データを結合
970     Water_point_all=[]
971     Water_times=[]
972     for i in range(len(water_level)):
973         Water_point=[water_level[i].data[i_t][0][i_y,i_x] for i_t in range(
974             water_level[i].times.size)]
975         Water_point_all=Water_point_all+Water_point
976         Water_times.append(water_level[i].times)
977     #時刻情報を生成
978     Water_times_all=np.concatenate(Water_times)
979     MBs=copy.deepcopy(water_level[0].MBs)
980     unit=copy.deepcopy(water_level[0].unit)
981     return FlowData(Water_point_all,Water_times_all,MBs,unit)
982
983 # In [52]:
984
985
986 #複数テキストデータを読み込んだ場合のリストのすべての地点をテキストデータごとリストではなくサンプリング時間ごとのリストにする。'''
987 #これを通すと次元限定の関数が使える2
988 def WaterTimeseries_point_all(water_level):
989     #ある座標の水面変化の時系列データを結合
990     Water_point_all=[]
991     Water_times=[]
992     for i in range(len(water_level)):
993         Water_point=[water_level[i].data[i_t] for i_t in range(water_level[i].
994             times.size)]
995         Water_point_all=Water_point_all+Water_point
996         Water_times.append(water_level[i].times)
997     #時刻情報を生成
998     Water_times_all=np.concatenate(Water_times)
999     MBs=copy.deepcopy(water_level[0].MBs)
1000     unit=copy.deepcopy(water_level[0].unit)
1001     return FlowData(Water_point_all,Water_times_all,MBs,unit)
1002
1003 # In [53]:
1004
```

```
1005
1006 #を通さないでもよい関数WaterTimeseries_point_all
1007 #関数の中を通してしているWaterTimeseries_point
1008 #これは例でこの関数のように2データ限定の関数を改造できる。D
1009 def PlotTimeseries_ix_2D_multiple(water_level,i_y=13, i_xmin=0, i_xmax=800, n_x
    =12, t_start=900,t_end=1000):
1010     fig = Plot(figsize=(10, 5))
1011     ax = fig.gca()
1012     di_x=int((i_xmax+1-i_xmin)/n_x)
1013
1014     for i_x in range(i_xmin,i_xmax+1,di_x):
1015         test1=WaterTimeseries_point(water_level,i_x,i_y)
1016         test1_time=TimeSeries(test1.data,t0=test1.t0,dt=test1.dt,unit=test1.unit)
1017         color = cm.viridis(float((i_x-i_xmin)/(i_xmax-i_xmin)))
1018         ax.plot(test1_time,color=color,label='x_{}_um,y_{}_um'.format(test1.
            MBs[0].x[i_x],test1.MBs[0].y[i_y]))
1019     ax.set_title('Water_level_Time_series')
1020     ax.set_xscale('auto-gps',epoch=0)
1021     ax.set_ylabel('[m]')
1022     ax.legend(bbox_to_anchor=(1.05, 1), loc='upper_left' )
1023     ax.set_xlim(t_start,t_end)
1024     return fig
1025
1026
1027 # In[54]:
1028
1029
1030 fig = Plot(figsize=(10, 5))
1031 ax = fig.gca()
1032
1033 fig2 = Plot(figsize=(10, 5))
1034 ax2 = fig2.gca()
1035 #for i_y in range(300):
1036
1037     t_start=900
1038     t_end=1000
1039     fftlength=10
1040
1041     i_x=-1
1042     i_y=13
1043     i_x_2=1425
1044     i_y_2=-201
1045     i_x_3=1425
1046     i_y_3=-209
1047     i_x_4=444
1048     i_y_4=-218
1049     i_x_5=502
1050     i_y_5=-276
1051
1052
```

```
1053
1054     test1=WaterTimeseries_point(water_level,i_x,i_y)
1055     test1_time=TimeSeries(test1.data,t0=test1.t0,dt=test1.dt,unit=test1.unit)
1056     test1_ASD=test1_time.crop(t_start,t_end).asd(fftlength,fftlength/2,method='Welch')
1057     ax.plot(test1_time,label='xu={:}m,yu={:}m'.format(test1.MBs[0].x[i_x],test1.
1058             MBs[0].y[i_y]))
1059
1060     '''
1061     test2=WaterTimeseries_point(water_level,i_x_2,i_y_2)
1062     test2_time=TimeSeries(test2.data,t0=test2.t0,dt=test2.dt,unit=test2.unit)
1063     test2_ASD=test2_time.crop(t_start,t_end).asd(fftlength,fftlength/2,method='Welch')
1064     ax.plot(test2_time,label='x = {:} m, y = {:} m'.format(test2.MBs[0].x[i_x_2],
1065             test2.MBs[0].y[i_y_2]))
1066     ax2.plot(test2_ASD,label='x = {:} m, y = {:} m'.format(test2.MBs[0].x[i_x_2],
1067             test2.MBs[0].y[i_y_2]))
1068
1069     test3=WaterTimeseries_point(water_level,i_x_3,i_y_3)
1070     test3_time=TimeSeries(test3.data,t0=test3.t0,dt=test3.dt,unit=test3.unit)
1071     test3_ASD=test3_time.crop(t_start,t_end).asd(fftlength,fftlength/2,method='Welch')
1072     ax.plot(test3_time,label='x = {:} m, y = {:} m'.format(test3.MBs[0].x[i_x_3],
1073             test3.MBs[0].y[i_y_3]))
1074     ax2.plot(test3_ASD,label='x = {:} m, y = {:} m'.format(test3.MBs[0].x[i_x_3],
1075             test3.MBs[0].y[i_y_3]))
1076
1077     test4=WaterTimeseries_point(water_level,i_x_4,i_y_4)
1078     test4_time=TimeSeries(test4.data,t0=test4.t0,dt=test4.dt,unit=test4.unit)
1079     test4_ASD=test4_time.crop(t_start,t_end).asd(fftlength,fftlength/2,method='Welch')
1080     ax.plot(test4_time,label='x = {:} m, y = {:} m'.format(test4.MBs[0].x[i_x_4],
1081             test4.MBs[0].y[i_y_4]))
1082     ax2.plot(test4_ASD,label='x = {:} m, y = {:} m'.format(test4.MBs[0].x[i_x_4],
1083             test4.MBs[0].y[i_y_4]))
1084
1085     test5=WaterTimeseries_point(water_level,i_x_5,i_y_5)
1086     test5_time=TimeSeries(test5.data,t0=test5.t0,dt=test5.dt,unit=test5.unit)
1087     test5_ASD=test5_time.crop(t_start,t_end).asd(fftlength,fftlength/2,method='Welch')
1088     ax.plot(test5_time,label='x = {:} m, y = {:} m'.format(test5.MBs[0].x[i_x_5],
1089             test5.MBs[0].y[i_y_5]))
1090     ax2.plot(test5_ASD,label='x = {:} m, y = {:} m'.format(test5.MBs[0].x[i_x_5],
1091             test5.MBs[0].y[i_y_5]))
1092
1093     '''
1094     ax.set_title('Water_level_Time_series')
1095     ax.set_xscale('auto-gps',epoch=0)
1096     ax.set_ylabel('Water_level[m]')
1097     ax.legend(bbox_to_anchor=(1.05, 1), loc='upper_left')
1098     ax.set_xlim(900,1000)
1099
1100     ax2.set_title('t={:}-{:}s'.format(t_start,t_end))
```

```
1093 ax2.set_xlabel('frequency [Hz]')
1094 ax2.set_ylabel('Water_level ASD [m/rtHz]')
1095 ax2.legend(bbox_to_anchor=(1.05, 1), loc='upper left')
1096 ax2.set_xscale('log')
1097 ax2.set_yscale('log')
1098
1099
1100 # In [55]:
1101
1102
1103 #を通す。WaterTimeseries_point_all
1104 #これで2データ限定の関数が見えるD
1105 New_water_level=WaterTimeseries_point_all(water_level)
1106
1107
1108 # In [56]:
1109
1110
1111 plot_1,TS=PlotTimeseries_ix_2D(New_water_level,i_y=13,i_xmin=0,i_xmax=800,n_x=12,
1112     t_start=900,t_end=1000)
1113
1114 # In [57]:
1115
1116
1117 plot_2,TS_ASD=PlotASDs(TS,fftlength=10,t_start=900,t_end=1000)
1118
1119
1120 # In [66]:
1121
1122
1123 PlotSpectrogram_x(New_water_level,i_y=13,t_start=900,t_end=1000,vmin=1e-9,vmax=1e
1124     -3,fftlength=10)
1125
1126 # In [77]:
1127
1128
1129 PlotSpectrogram_Coherence(New_water_level,i_y=13,i_x_0=600,t_start=900,t_end=1000,
1130     fftlength=10)
1131
1132 # In [59]:
1133
1134
1135 PlotCoherence(TS,6,fftlength=10,t_start=900,t_end=1000)
1136
1137
1138 # In [61]:
1139
```

```
1140
1141 #以下を利用したものWatertimeseries_point例の関数()面倒で関数にはしていません()
1142
1143
1144 # In [62]:
1145
1146
1147 fig = Plot(figsize=(10, 5))
1148 ax2 = fig.gca()
1149 i_xmin=0
1150 i_xmax=800
1151 n_x=12
1152 i_y=13
1153 di_x=int((i_xmax+1-i_xmin)/n_x)
1154 t_start=900
1155 t_end=1000
1156 fftlength=10
1157
1158 for i_x in range(i_xmin,i_xmax+1,di_x):
1159     test1=WaterTimeseries_point(water_level,i_x,i_y)
1160     test1_time=TimeSeries(test1.data,t0=test1.t0,dt=test1.dt,unit=test1.unit).crop
        (t_start,t_end).asd(fftlength,fftlength/2,method='Welch')
1161     color = cm.viridis(float((i_x-i_xmin)/(i_xmax-i_xmin)))
1162     ax2.plot(test1_time,color=color,label='x={:}m,y={:}m'.format( test1.MBs
        [0].x[i_x],test1.MBs[0].y[i_y]))
1163
1164
1165 ax2.set_title('t={:}-{:}s'.format(t_start,t_end))
1166 ax2.set_xlabel('frequency[Hz]')
1167 ax2.set_ylabel('Water_level_ASD[m/rtHz]')
1168 ax2.legend(bbox_to_anchor=(1.05, 1), loc='upper_left' )
1169 ax2.set_xscale('log')
1170 ax2.set_yscale('log')
1171
1172
1173 # In [63]:
1174
1175
1176 i_y=0
1177 i_x_min=0
1178 i_x_max=800
1179 t_start=900
1180 t_end=1000
1181 fftlength=10
1182
1183 asd=[TimeSeries(WaterTimeseries_point(water_level,i_x,i_y).data,
1184             t0=WaterTimeseries_point(water_level,i_x,i_y).t0,
1185             dt=WaterTimeseries_point(water_level,i_x,i_y).dt,
1186             unit=WaterTimeseries_point(water_level,i_x,i_y).unit).crop(t_start
        ,t_end).asd(fftlength,fftlength/2,method='Welch') for i_x in
```

```

                                range(i_x_min,i_x_max+1)]
1187 ASD_xf=np.vstack([a.value for a in asd]).T
1188 fig = Plot(figsize=(15, 5))
1189 ax = fig.gca()
1190 ax.set_xlabel("x [m]")
1191 ax.set_ylabel("f [Hz]")
1192 ax.set_yscale('log')
1193 ax.set_ylim(0.1,10)
1194 ax.set_title('ASD_{y}_{x}={:}_{m},t={:}-{:}s'.format( WaterTimeseries_point(water_level
    ,0,0).MBs[0].y[i_y],t_start,t_end))
1195 im=ax.imshow(ASD_xf,
1196             extent=[WaterTimeseries_point(water_level,0,0).MBs[0].x[i_x_min],
1197                   WaterTimeseries_point(water_level,0,0).MBs[0].x[i_x_max],
1198                   asd[0].frequencies[0].value,
1199                   asd[0].frequencies[-1].value],
1200             norm='log', vmin=1e-8, vmax=1e-3, aspect='auto', origin='lower',
1201             interpolation='nearest')
1202 fig.colorbar(im, ax=ax)
1203
1204 # In [64]:
1205
1206
1207 i_x=-1
1208 i_y_min=0
1209 i_y_max=27
1210 t_start=900
1211 t_end=1000
1212 fftlength=10
1213
1214 asd=[TimeSeries(WaterTimeseries_point(water_level,i_x,i_y).data,
1215             t0=WaterTimeseries_point(water_level,i_x,i_y).t0,
1216             dt=WaterTimeseries_point(water_level,i_x,i_y).dt,
1217             unit=WaterTimeseries_point(water_level,i_x,i_y).unit).crop(t_start
    ,t_end).asd(fftlength,fftlength/2,method='Welch') for i_y in
    range(i_y_min,i_y_max+1)]
1218 ASD_yf=np.vstack([a.value for a in asd]).T
1219 fig = Plot(figsize=(15, 5))
1220 ax = fig.gca()
1221 ax.set_xlabel("y [m]")
1222 ax.set_ylabel("f [Hz]")
1223 ax.set_yscale('log')
1224 ax.set_ylim(0.1,10)
1225 ax.set_title('ASD_{x}_{y}={:}_{m},t={:}-{:}s'.format( WaterTimeseries_point(water_level
    ,0,0).MBs[0].x[i_x],t_start,t_end))
1226 im=ax.imshow(ASD_yf,
1227             extent=[WaterTimeseries_point(water_level,0,0).MBs[0].y[i_y_min],
1228                   WaterTimeseries_point(water_level,0,0).MBs[0].y[i_y_max],
1229                   asd[0].frequencies[0].value,
1230                   asd[0].frequencies[-1].value],
```

```
1231         norm='log', vmin=1e-8, vmax=1e-3, aspect='auto', origin='lower',  
        interpolation='nearest')  
1232     fig.colorbar(im, ax=ax)
```

## 参考文献

- [1] 佐藤勝彦, ”相対性理論 (15 版)”, 岩波書店、2013 年
- [2] LIGO 公式ホームページ、<https://www.ligo.caltech.edu/>
- [3] Virgo 公式ホームページ、<https://www.virgo-gw.eu/>
- [4] GEO600 公式ホームページ、<https://www.geo600.org/>
- [5] KAGRA 公式ホームページ、<https://gwcenter.icrr.u-tokyo.ac.jp/plan>
- [6] Einstein Telescope (ET) official web site, <https://www.et-gw.eu/index.php>
- [7] LISA Consortium, <https://www.elisascience.org/>
- [8] T. Akutsu et al. (KAGRA Collaboration), ”Overview of KAGRA: Detector design and construction history”, 2021, Prog. Theor. Exp. Phys., 05A101 , <https://doi.org/10.1093/ptep/ptaa125>
- [9] 中村卓史・三尾典克・大橋正健編著、重力波をとらえる-存在の証明から検出へ、京都大学学術出版会、1998
- [10] H.Abe et al. (KAGRA Collaboration), ”Performance of the KAGRA detector during the first joint observation with GEO600 (O3GK)”, 2022, Prog. Theor. Exp. Phys., ptac093 , <https://doi.org/10.1093/ptep/ptac093>
- [11] Rishabh Bajpai et al. , ”Estimation of Newtonian noise from KAGRA cooling system” , 2022 , arXiv:2207.13466, <https://doi.org/10.48550/arXiv.2207.13466>
- [12] A Nishizawa, internal paper , JGW-T1706440-v2, Water Newtonian noise for KAGRA , <https://gwdoc.icrr.u-tokyo.ac.jp/cgi-bin/private/DocDB/ShowDocument?docid=6440>
- [13] K.somiya, Newtonian noise from the underground water, internal paper , <https://gwdoc.icrr.u-tokyo.ac.jp/cgi-bin/private/DocDB/ShowDocument?docid=10792>
- [14] Y. Chen, “Gravity Gradient Noise from Water” , internal document (2016), .
- [15] 株式会社クリモトポリマーホームページ、製品紹介、[https://www.kuripoly.jp/product/kurimoto\\_press/](https://www.kuripoly.jp/product/kurimoto_press/)
- [16] 宮井善弘・木田輝彦・仲谷仁志・巻幡敏秋著、”水力学 (第二版)”、森北出版株式会社、2014
- [17] 大津岩夫、他、”水理学”、理工図書株式会社、2007
- [18] 岡本芳美、”実用水理学ハンドブック”、築地書館株式会社、2016
- [19] 中矢哲郎、他、”摩耗したコンクリート水路表層形状からの粗度係数推定手法”、2008、農業農村工学会論文集 (2008 巻、258 号、P.501-506、a1)、<https://doi.org/10.11408/jsidre2007.2008.501>
- [20] 加藤敬、他、”開水路における壁面の凹凸から水路の粗度係数を求める試み”、2008、農村工学研究所技報 207 号 (P.183-193)、[https://www.naro.go.jp/PUBLICITY\\_REPORT/publication/archive/files/207-14.pdf](https://www.naro.go.jp/PUBLICITY_REPORT/publication/archive/files/207-14.pdf)
- [21] 測定キューブホームページ、表面輪郭ゲージ SRT-6223(日本ソナテスト) 、[https://sokutei-cube.com/products/shosai/2312?gclid=Cj0KCQjwjo2JBhCRARIsAFG667XirEhRtdB\\_sUa47wclrvyBgFUyD81kUnWs10Lnv0SYD5n6snMVxjMaAsa0EALw\\_wcB#yoto](https://sokutei-cube.com/products/shosai/2312?gclid=Cj0KCQjwjo2JBhCRARIsAFG667XirEhRtdB_sUa47wclrvyBgFUyD81kUnWs10Lnv0SYD5n6snMVxjMaAsa0EALw_wcB#yoto)
- [22] Flow science 社ホームページ、<https://www.flow3d.co.jp/>
- [23] GWpy documentation(Python package)、<https://gwpy.github.io/docs/stable/index.html>

- 
- [24] Creo Elements Direct/Modeling Express 8.0 ダウンロードページ、<https://www.ptc.com/ja/products/creo/elements-direct/modeling-express>, Parametric Technology Corporation 提供
- [25] A Nishizawa, internal paper, JGW-T2214308-v1, Edge contribution of a water channel to a Newtonian noise, <https://gwdoc.icrr.u-tokyo.ac.jp/cgi-bin/private/DocDB/ShowDocument?docid=14308>
- [26] T. Akutsu et al., "First cryogenic test operation of an underground km-scale gravitational-wave observatory KAGRA", 2019, *Class. Quant Grav.* 36 165008 <https://iopscience.iop.org/article/10.1088/1361-6382/ab28a9>
- [27] 井上崇著, 重力波望遠鏡における地下水の重力勾配雑音の研究, 宗宮研卒論, 2018

# 謝辞

本研究及び、本修士論文執筆にあたり、多くの方から協力をいただきました。まず、私が宗宮研に所属した学部4年時より本修士論文執筆まで3年間ご指導をいただきました指導教官の宗宮健太郎先生に感謝申し上げます。宗宮先生には私が本研究テーマを始めた当初より、先行研究の説明や、結果に関する議論など様々な助言をいただきました。さらに私が研究室に入った同時期から新型コロナウイルスの世界的流行という困難な状況にありましたがこの修士論文を執筆する今日まで研究活動を行えるように研究環境を整えるなど支援いただきました。また、KAGRAへの出張や修士過程では多くの学会研究会で研究成果を発表する機会をいただくなど研究活動の支援をいただきました。特に、人の移動が制限される状況の中で2020年11月と2021年8月にKAGRAへ訪問することができたのは、宗宮先生や出張のルールを定めてくださった東京工業大学、受け入れ先であるKAGRAの協力があってからだと感じています。

本研究テーマを研究する上で、週1のKAGRAのPEMグループのミーティングに参加し、そこで本研究について様々な議論をさせていただきました。KAGRAのPEMグループのメンバーの方々には感謝申し上げます。特に、国立天文台重力波プロジェクトの鷺見貴生さん、東京大学宇宙線研究所の横澤孝章さん東京大学大学院理学系研究科の西澤篤志さんには多くの議論や研究協力をいただきました。具体的に西澤さんには本研究における理論面の議論、横澤さんには研究に関するアドバイスやKAGRA訪問中の案内、PEMとして研究発表を行う際に議論やコメントをいただき、研究を進めることができました。そして鷺見さんには多くの協力をしていただき全ての協力をここに記すことはできませんがいくつかの貢献を挙げて感謝申し上げたいと思います。まず、本研究において重要なKAGRAにおける水の流量や流速などを実際に測定器を導入し、計測を行っていただきました。この計測によるデータや水の様子は本研究を進める上で重要なものになりました。次に、私自身に関しても2021年の8月にKAGRAへ10日間ほど滞在させてもらった時に、本論文のシミュレーションの解析に必要なプログラムを作成する際に、プログラムを見ていただき、多くのアドバイスをいただきました。この出張中は他にもKAGRAのパイプの状況など解説していただき、本論文の研究を進める上で大変重要な滞在とすることができました。研究とは関係ないことですが、富山市内のたくさんの飲食店で時に奢っていただき食事をご一緒でき楽しい滞在となりました。他にも私がいる東工大へ訪問していただき、Y-endを再現したパイプを設計するためのCADソフトの使い方を教えていただくなど、本研究を進めるためのスキルを私に教えて下さり、大変感謝しています。

研究室では私が学部4年時に本研究テーマに取り組む前に、Flow-3Dによるシミュレーションを行っていた先輩のLiu YutingさんにFlow-3Dの操作を教えてくださいました。さらにFlow-3Dに関してはFlow science社日本法人のサポート担当の方と技術者の方にメールやミーティングを通じて詳しく仕様についてご教示いただきました。また、本研究を始める前に本研究に取り組んでいらした先輩の井上崇さんの卒論を参考に前提知識などを勉強させていただきました。他にも、私が宗宮研に在籍している間に宗宮研の先輩方には学部4年のときより輪講や研究室での実験で重力波に関することや他の多くのことを学ばせていただきましたことをここで感謝します。先輩方に限らず、後輩や同期である鈴木海堂君や原田健一先生にもゼミや輪講で議論したり、お世話になりました。

最後に、新型コロナウイルスの影響で研究もプライベートも対面でなかなか人と会えない中関わりを持ち続けてくれた友人と、私の生活を直接支えてくれた家族に感謝して謝辞としたいと思います。