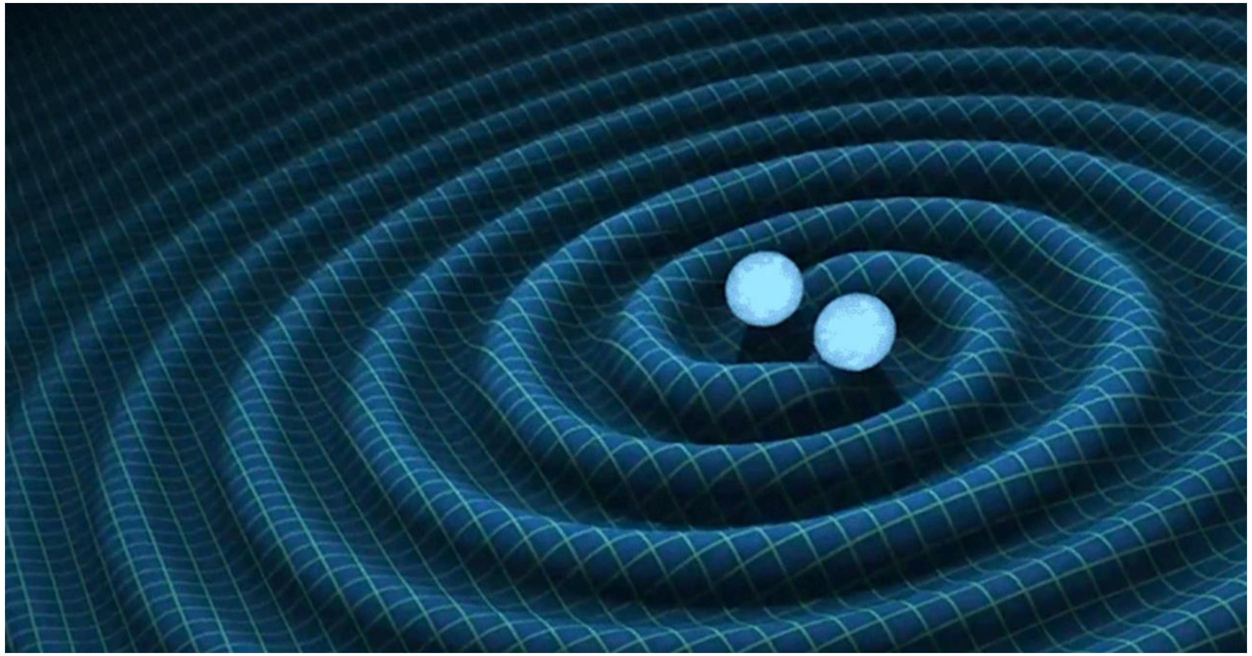
# 重力波望遠鏡KAGRAにおける地下水の重力勾配雑音

2020年12月16日　卒論中間発表会・先端研究基礎発表会

宗宮研究室

鈴木孝典

# 重力波とは



重力波のイメージ

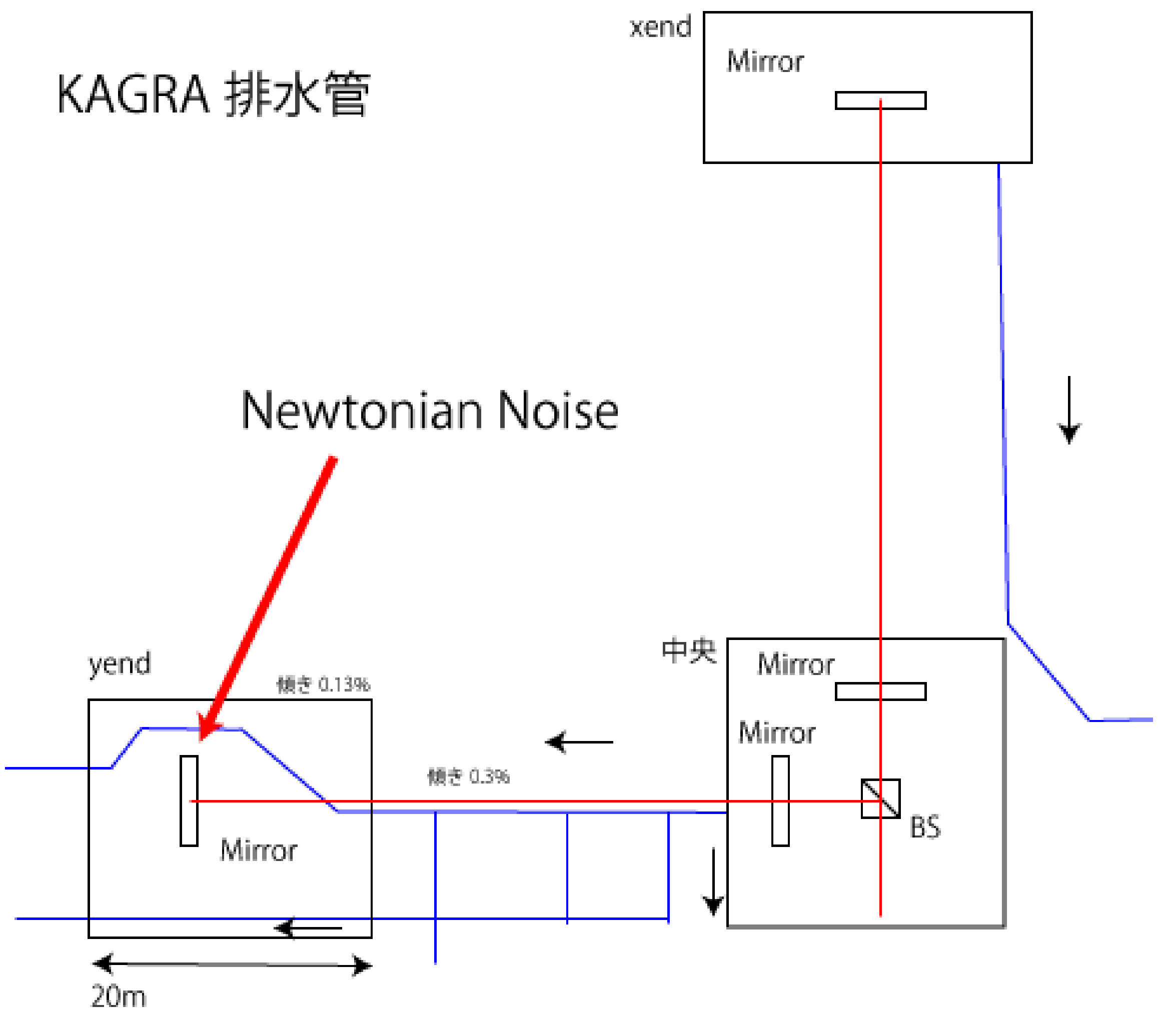
大型低温重力波望遠鏡KAGRAの俯瞰図

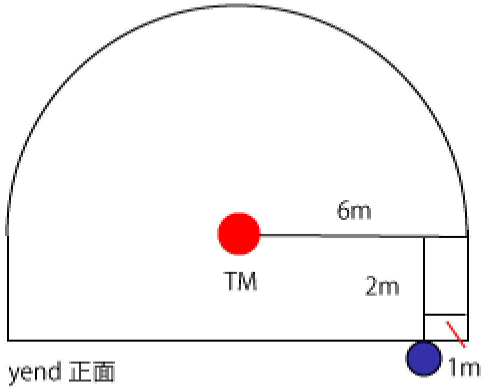
**画像提供：東京大学宇宙線研究所 重力波観測研究施設**

# 背景

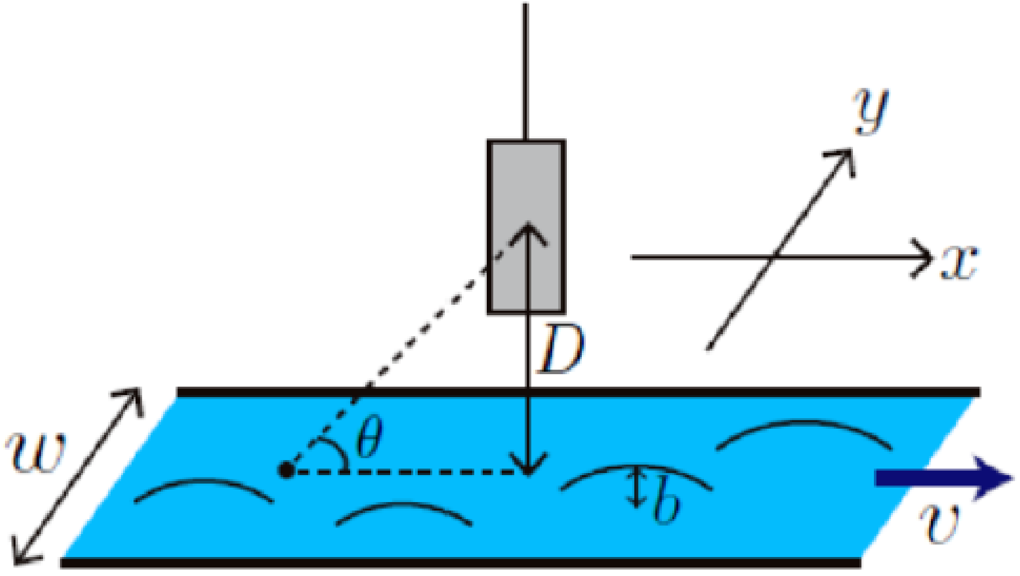
- 重力波望遠鏡KAGRAは地面振動を下げるために地下300mに建てられている。この地下では地下水がパイプを通って流れており、地下水が波打つことでKAGRAの鏡に対して重力勾配が生じ雑音となる。
- その雑音をパイプの流水をシミュレーションして計算する。

# 地下水を流すパイプ


KAGRA 排水管
Newtonian Noise
xend
Mirror
中央
Mirror
Mirror
BS
yend
傾き 0.13%
傾き 0.3%
Mirror
20m


6m
TM
2m
1m
yend 正面

# 重力勾配雑音の定式化



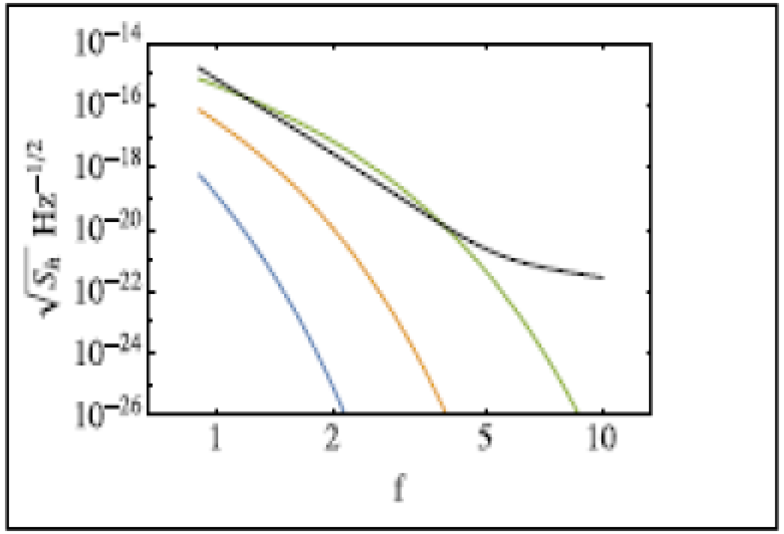

*a*:ミラーに与える加速度勾配(万有引力による雑音)

$$
\begin{aligned}
a &= \int\int \frac{G\rho b(x,t)\cos\theta}{D^2+x^2+y^2}dxdy \\
&= G\rho \int\int \frac{b(x,t)x}{(D^2+x^2+y^2)^{3/2}}dxdy \\
&= -G\rho \int\int \frac{\partial b(x,t)}{\partial x}\frac{1}{(D^2+x^2+y^2)^{1/2}}dxdy \\
&\approx -G\omega\rho \int \frac{\partial b(x,t)}{\partial x}\frac{1}{(D^2+x^2)^{1/2}}dx
\end{aligned}
$$

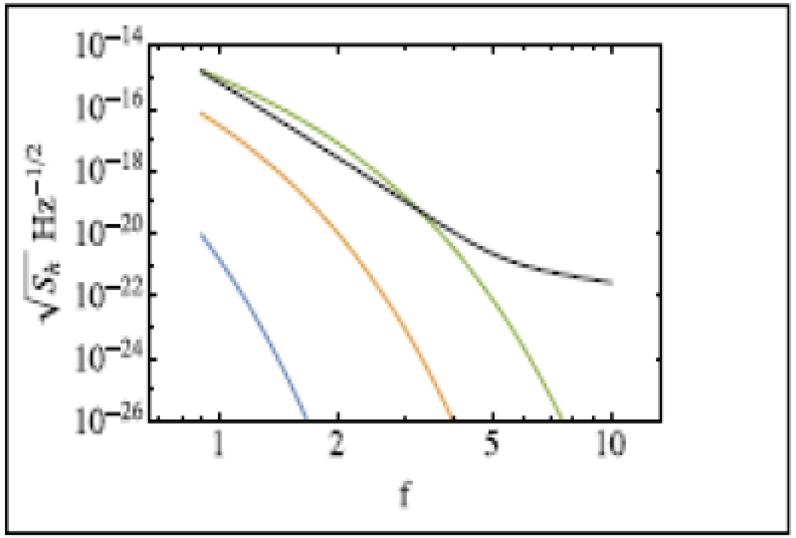
⇨ 水面の変動*b(x, t)*をシミュレーションする

# 西澤モデル

- 波の位相速度が一定、乱流が一様



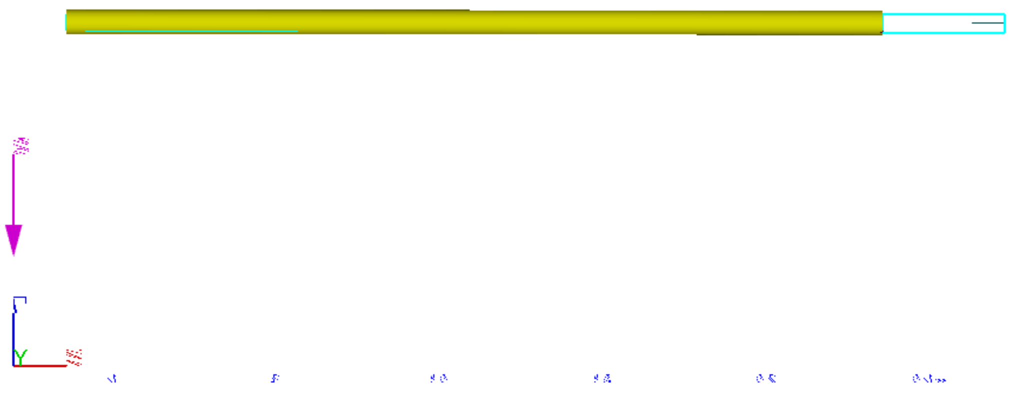

各水流の速さにおける雑音 $v = 1$m/s(青),$v = 2$(オレンジ),$v = 5$(緑),KAGRAsesitivity(black)




6.3　各 D の違いによる雑音 D=1m(緑),D=2m(オレンジ),D=5m(青)

# パイプの設計





| | | |
|---|---|---|
| Static friction coefficient | 0.16 | kg/m^2/s |
| Roughness height | 0.0016 | m |

Gravity and non-inertial reference frame

◉ Activate gravity

Gravity components

| | | |
|---|---|---|
| X component | 0.0294 | m/s^2 |
| Y component | 0 | m/s^2 |
| Z component | -9.702 | m/s^2 |

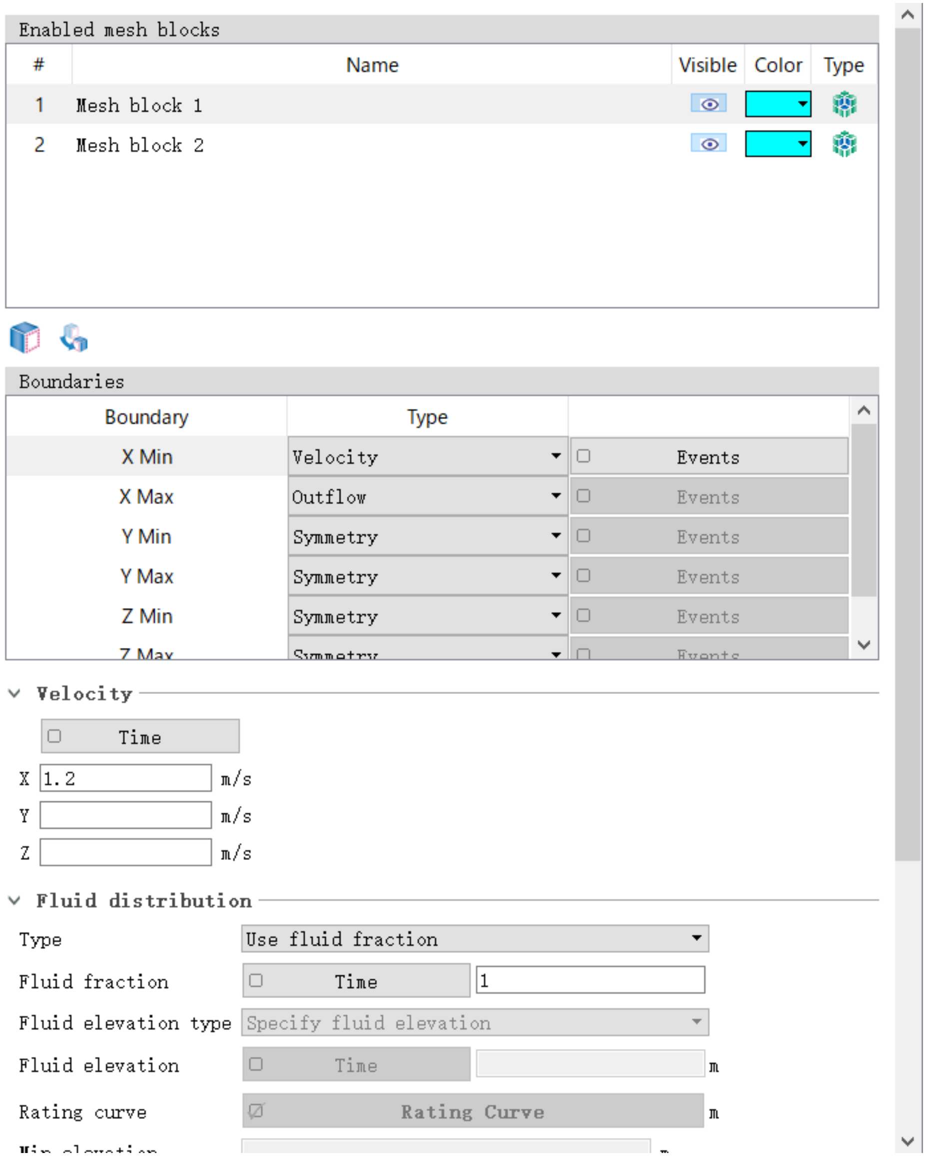
## パイプの概形

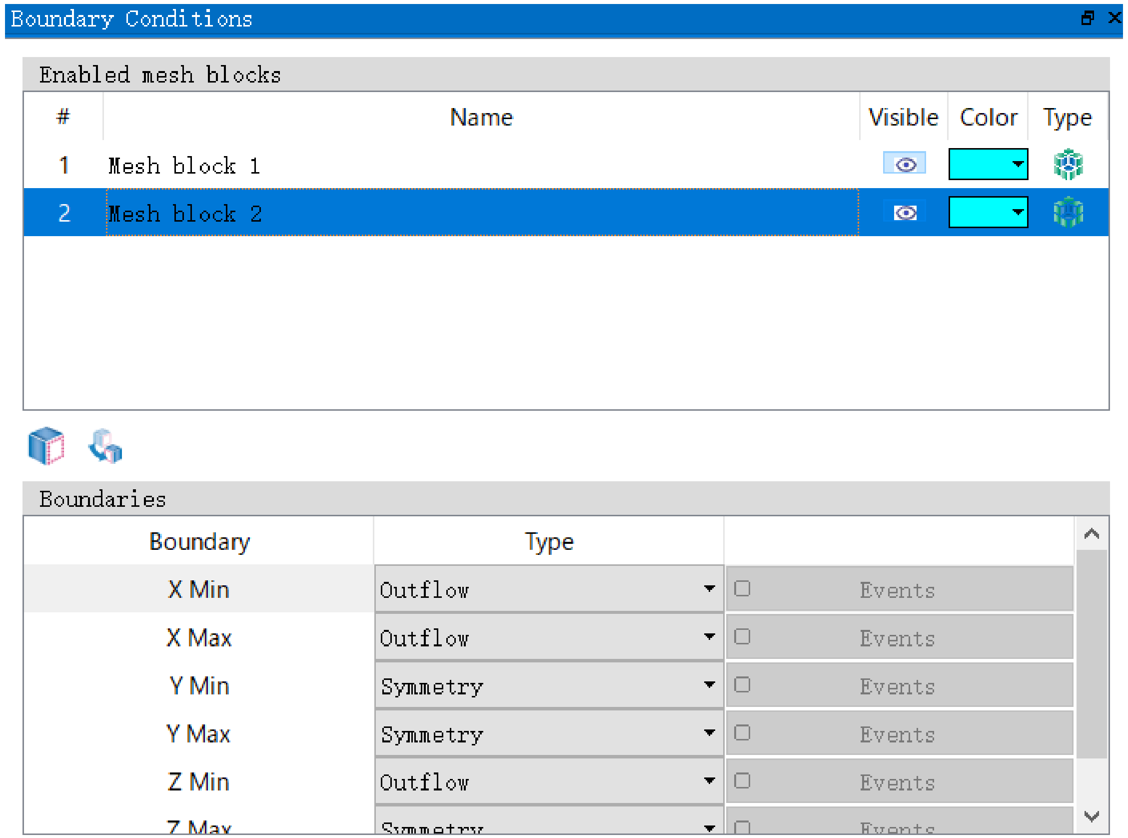
パイプ長:20m
半径:0.2m
表面粗さ:1.6mm
粗度係数0.016

重力加速度を成分で設定してパイプに傾斜を作る

傾斜:0.3%、0.13%でシミュレーションを行う

# 境界条件

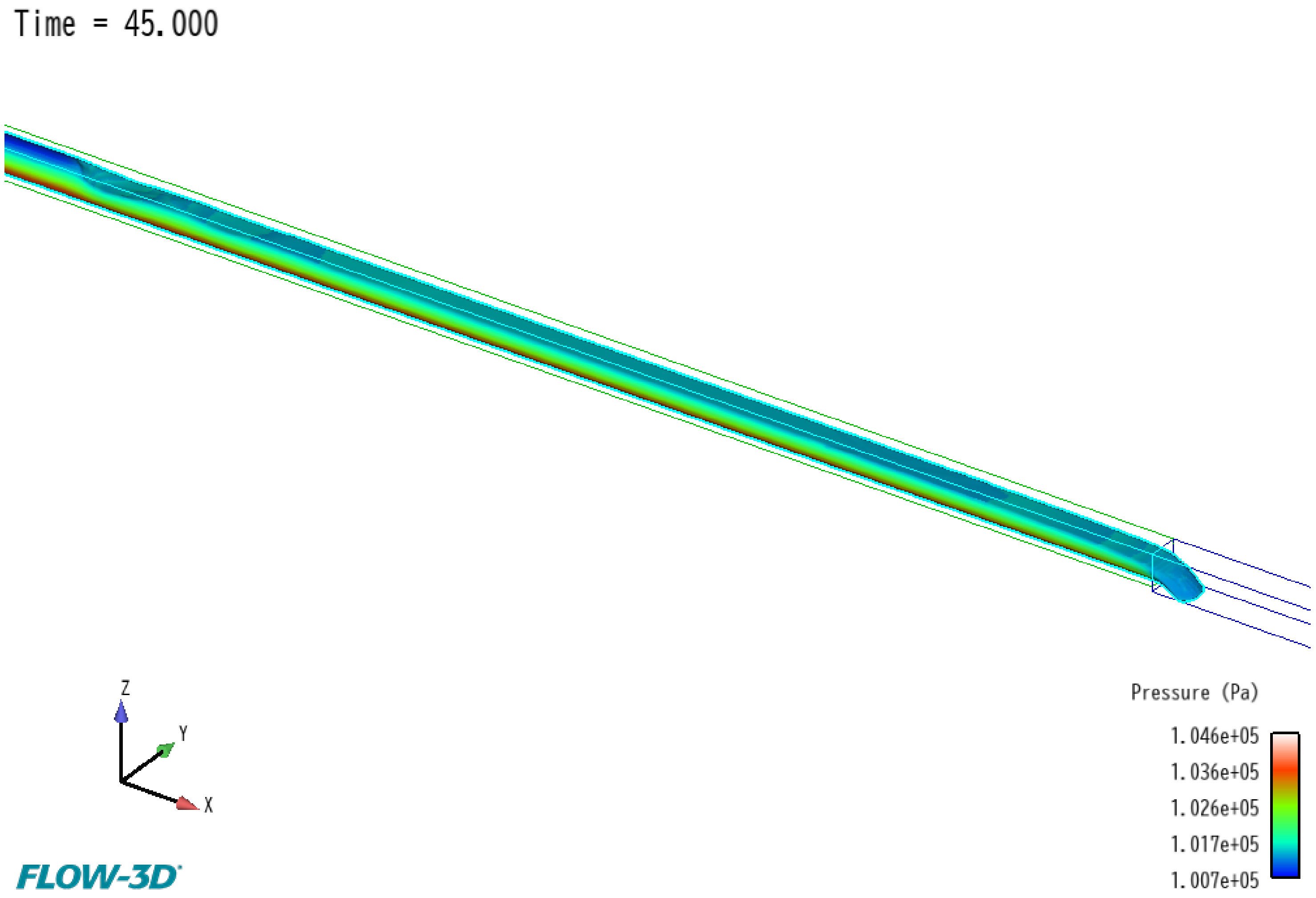



初速度
v=2m/s,1.2m/s,0.8m/s
でシミュレーション
(1.2m/sTilt0.13はシミュレーション中)

# シミュレーションの結果

- 水上自由サーフェイスの様子



# 重力勾配雑音の導出のプログラム1(MATLAB)

```
G=6.67408/100000000000; %万有引力定数
rho=997; %水の密度
Alldatalen=1018325;%(テキストの行数を手動入力)
r=0.2;%パイプの半径(手動入力)
cellsize=0.02^2;%cellの大きさ(面積)

%データ自体の個数cを求める。(ある時間のデータを1個とする)
text=fileread('waternoise.txt');%ファイルをテキストとして読む
expr = '[^¥n]*Mesh Block    1[^¥n]*';%検索するワードの定義
matches = regexp(text,expr,'match');%検索したワードが配列に格納
c=numel(matches);%配列の要素数を数える(検索したワードの数)

datalen=(Alldatalen/c)-5;%ある時間における生データの長さ
opts = detectImportOptions('waternoise.txt'); %インポートオプションの設定

%データを取るxの長さ(L)を求める
opts.DataLines = [6 (datalen+5)]; %データ範囲の指定
opts.VariableNames = {'x','y','z','hyfd'};%変数の命名
T0 = readtable('waternoise.txt',opts); %データのtableの作成
T=table2array(T0);%tableを配列に変換
%変数を配列に格納
x=T(:,1);
L=x(end,1)-x(1,1);%データxの最後から最初を引く

%ある地点での水深の時間平均
hyfdt=NaN(datalen,c);%空の配列
for h=1:c
opts.DataLines = [datalen*(h-1)+5*h+1 (datalen+5)*h]; %データ範囲の指定(6580(i-1)+5i+1)
opts.VariableNames = {'x','y','z','hyfd'};%変数の命名
T0 = readtable('waternoise.txt',opts); %データのtableの作成
T=table2array(T0)%tableを配列に変換
%水面となっていない地点を除外したnewTを作成
hyfdnot0=abs(T(:,4))>0;%水面となっていない地点を0


%変数を配列に格納
y=T(:,2);
hyfd0=T(:,4);%flow3Dの意味でのhyfd
hyfd=hyfd0+r-(r^2-y.^2).^0.5;%一番下からの水面の距離
hyfdt(:,h)=hyfd.*hyfdnot0;%縦方向に各地点の高さを入れる(横は時間)
end
hyfdt2=standardizeMissing(hyfdt,0);%hyfd内の0値をNaNに変更
M=nanmean(hyfdt2,'all');%水深の時間空間平均


for D=1:10
%ニュートンニアンノイズの配列を作成(時間)
a=zeros(1,c) %空の配列の作成
for i=1:c
opts.DataLines = [datalen*(i-1)+5*i+1 (datalen+5)*i]; %データ範囲の指定(6580(i-1)+5i+1)
opts.VariableNames = {'x','y','z','hyfd'};%変数の命名
T0 = readtable('waternoise.txt',opts); %データのtableの作成
T=table2array(T0)%tableを配列に変換
%水面となっていない地点を除外したnewTを作成
hyfdnot0=abs(T(:,4))>0;%水面となっていない地点を0
n1=numel(hyfdnot0);
k0=sum(hyfdnot0);
k1=1;
newT=zeros(k0,4);
for k=1:n1
if hyfdnot0(k,1)==1
    newT(k1,:)=T(k,:);
    k1=k1+1;
end
end

%変数を配列に格納
x=newT(:,1);
```

# 重力勾配雑音の導出のプログラム2

```
x=newT(:,1);
y=newT(:,2);
z=newT(:,3);
hyfd0=newT(:,4);%flow3Dの意味でのhyfd
hyfd=hyfd0+r-(r^2-y.^2).^0.5;%一番下からの水面の距離
y=newT(:,2);%y座標平行移動用の座標

b=hyfd-M; %平均からの差
n=numel(x);%xの要素数

F=G*rho*cellsize.*(x.*b)./((D^2+x.^2+y.^2).^1.5); %被積分関数

%Fを積分可能な行列に変換
%c1の導出
x1= x==x(1,1);
c1=sum(x1);
sdatalen=n/c1;%sdatalenの導出
F2=zeros(c1,sdatalen);%空の配列の作成
y1=zeros(c1,1);
for j=1:c1
    y1(j,1)=y(1+sdatalen*(j-1),1);%積分に必要なyの範囲を抽出
    F2(j,:)=F(1+sdatalen*(j-1):sdatalen*j,1).';%値をmeshgrid型に再配置
end

Q=trapz(y1,trapz(x(1:sdatalen,1).',F2,2));%traqz(積分変数,F,データの次元)
a(1,i)=Q;%ニュートンニアンノイズ

end

%ニュートンニアンノイズの時間依存
t0=0:c-1;
fs=20;%サンプリング周波数(手動入力)
t=(1/fs)*t0;%横軸の設定(時間)
figure(1);
```


```
for j=1:c1
    y1(j,1)=y(1+sdatalen*(j-1),1);%積分に必要なyの範囲を抽出
    F2(j,:)=F(1+sdatalen*(j-1):sdatalen*j,1).';%値をmeshgrid型に再配置
end

Q=trapz(y1,trapz(x(1:sdatalen,1).',F2,2));%traqz(積分変数,F,データの次元)
a(1,i)=Q;%ニュートンニアンノイズ

end

%ニュートンニアンノイズの時間依存
t0=0:c-1;
fs=20;%サンプリング周波数(手動入力)
t=(1/fs)*t0;%横軸の設定(時間)
figure(1);
```

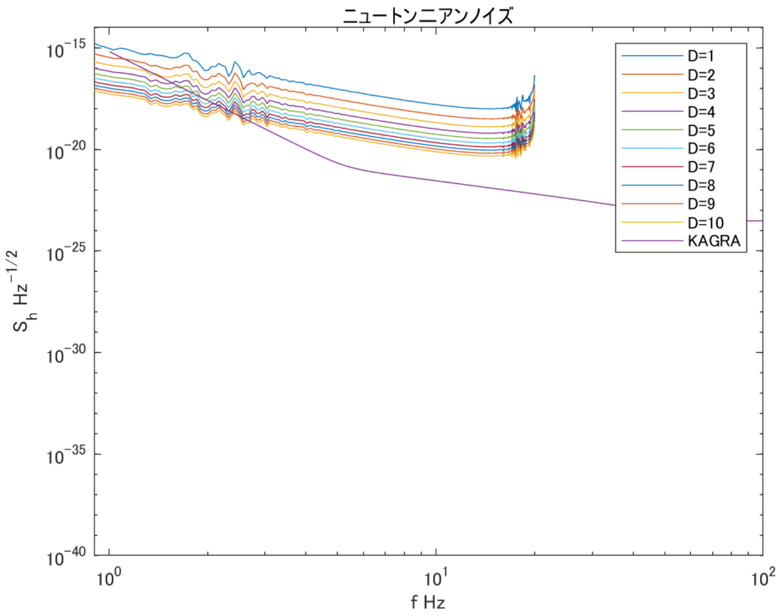

```
plot(t,a);
xlabel('t s')
ylabel('Newtoniannoise')
title('ニュートンニアンノイズの時間依存性')
hold on

%パワースペクトルの導出
aw=abs(fft(a));%sqrt(パワースペクトル密度)
f=1:c;
f=f*fs/c;%横軸の設定(周波数)
sh=aw./(4*pi^2.*f.^2);%ひずみの次元への修正
figure(2);
loglog(f,sh);
xlabel('f Hz')
ylabel('S_h Hz^{-1/2}')
title('ニュートンニアンノイズ')
hold on
end
```

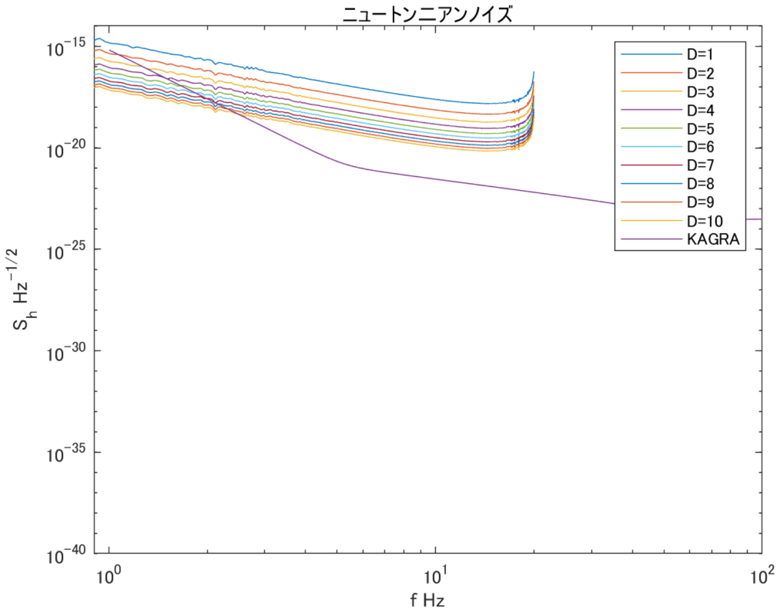
# 重力勾配雑音の計算結果1

- 傾き0.3%、パイプの真上にミラーがある場合

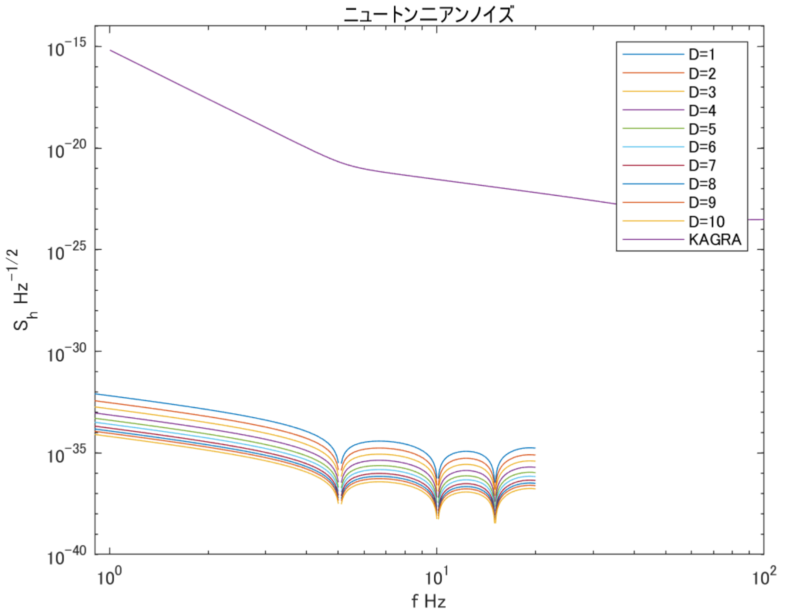
D=1−10m Time25s−45s −2m<x<2m




V=0.8m/s




V=1.2m/s

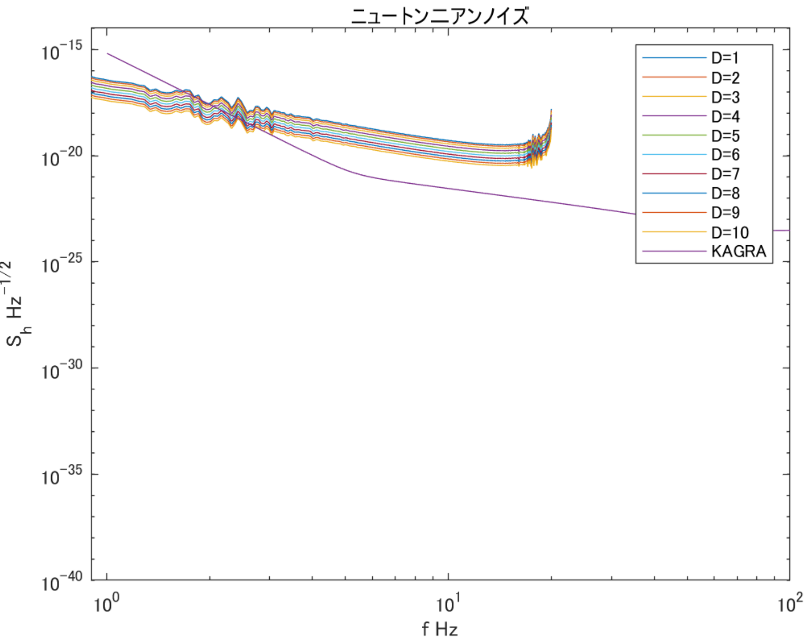



V=2.0m/s(パイプがほぼ満水)

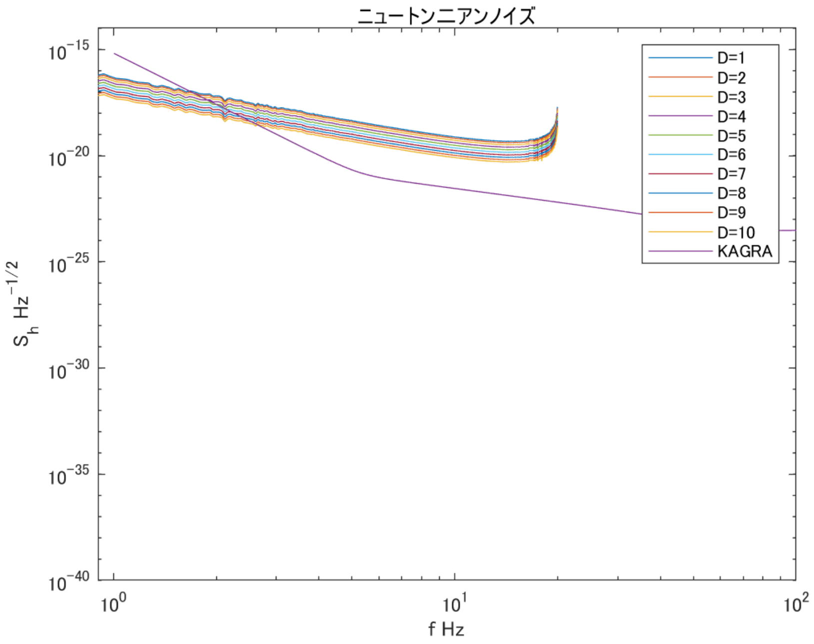
# 重力勾配雑音の計算結果2(より現実の系に近い)

- 傾き0.3%、パイプが5m横にある場合

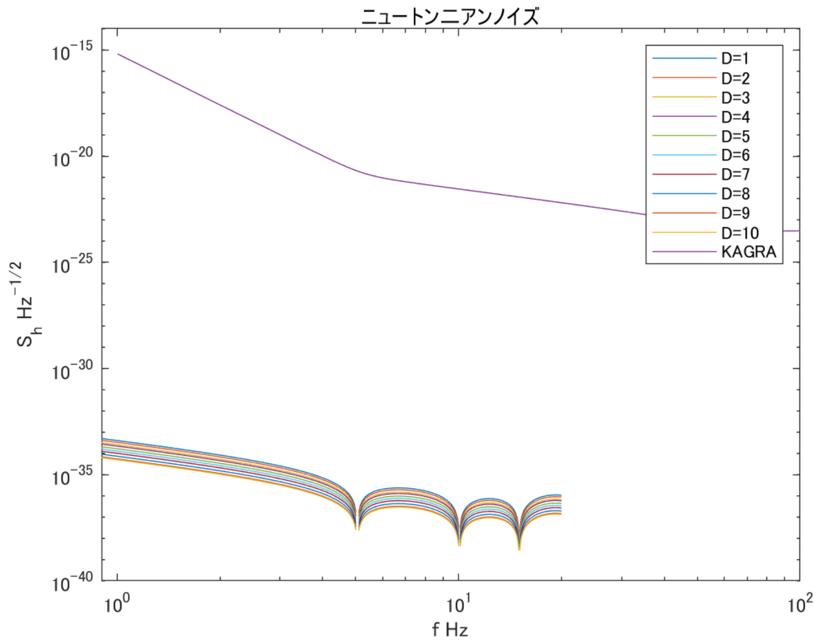
D=1-10m Time25s-45s -2m<x<2m




V=0.8m/s




V=1.2m/s



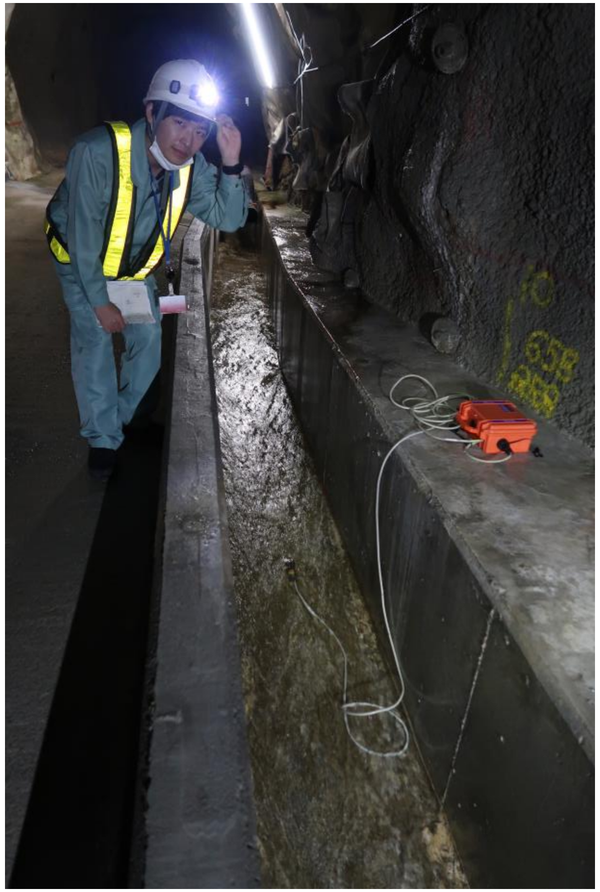

V=2.0m/s (パイプがほぼ満水)

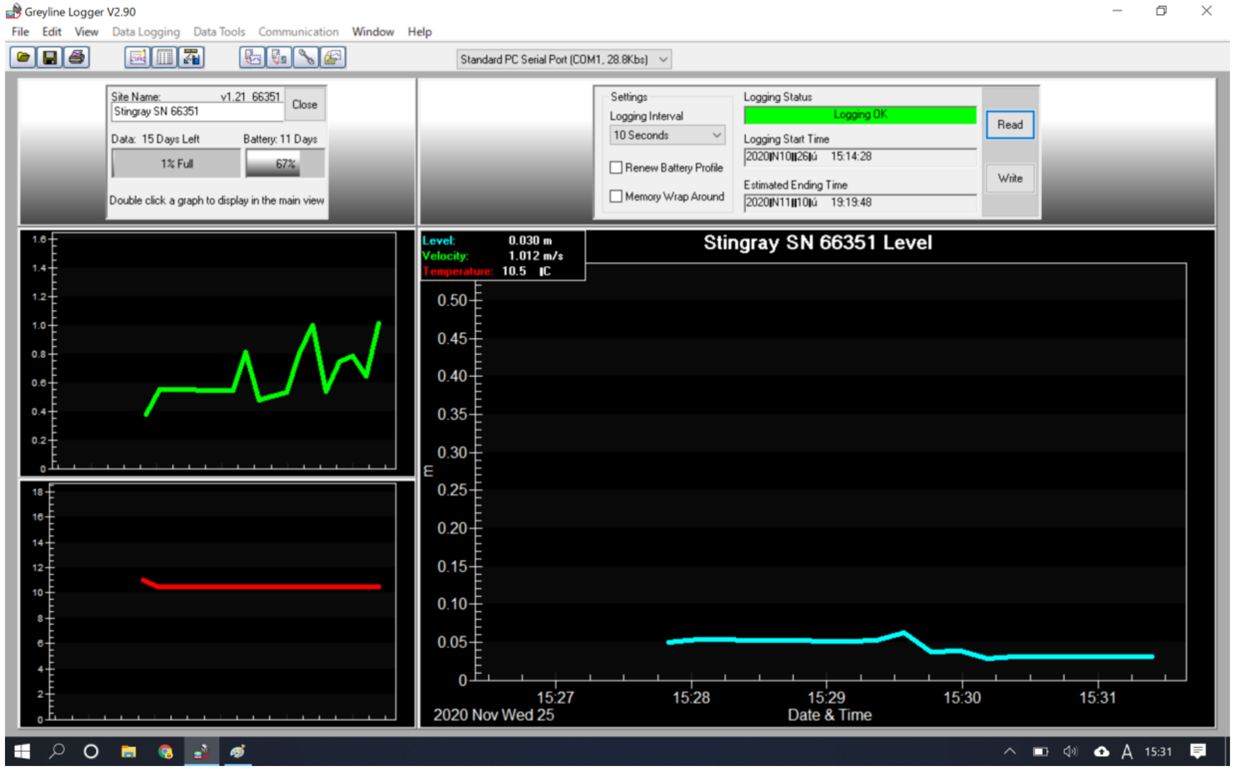
# まとめ

Tokyo Tech

- 基本的にパイプと鏡の距離が遠くなるほど雑音は小さくなる。
- パイプ全体に水が通っているなど水面が十分に落ち着いていればKAGRAの目標感度には雑音として現れない。
- V=0.8,1.2m/sの時(水深がパイプの半分)には、水面が動きやすいためKAGRAの目標感度を制限する可能性がある。

# 課題：実際の水量






簡易水量計

実際の水量データ(中央排水管+α)

# 課題・今後の方針

- 傾き0.3%のものはシミュレーション時間が45sのため状態が落ち着いていない可能性がある。

→シミュレーション時間を増やす。

- 水面の平均の取り方を工夫する。
- 傾き0.13%のシミュレーションを終わらせる。
- プログラムの見直し。

Tokyo Tech

# ご清聴ありがとうございました