可視化の基礎演習

担当教員　大野

概要

- 3次元ベクトルデータの可視化
  - OpenDXの起動
  - Generalファイル作成
  - Visualization Networkの例
    - 流線(Streamline, Tube), 矢印
  - フォルダはdx_11th。デスクトップにコピーすること。MyDocumentにはコピーしないこと

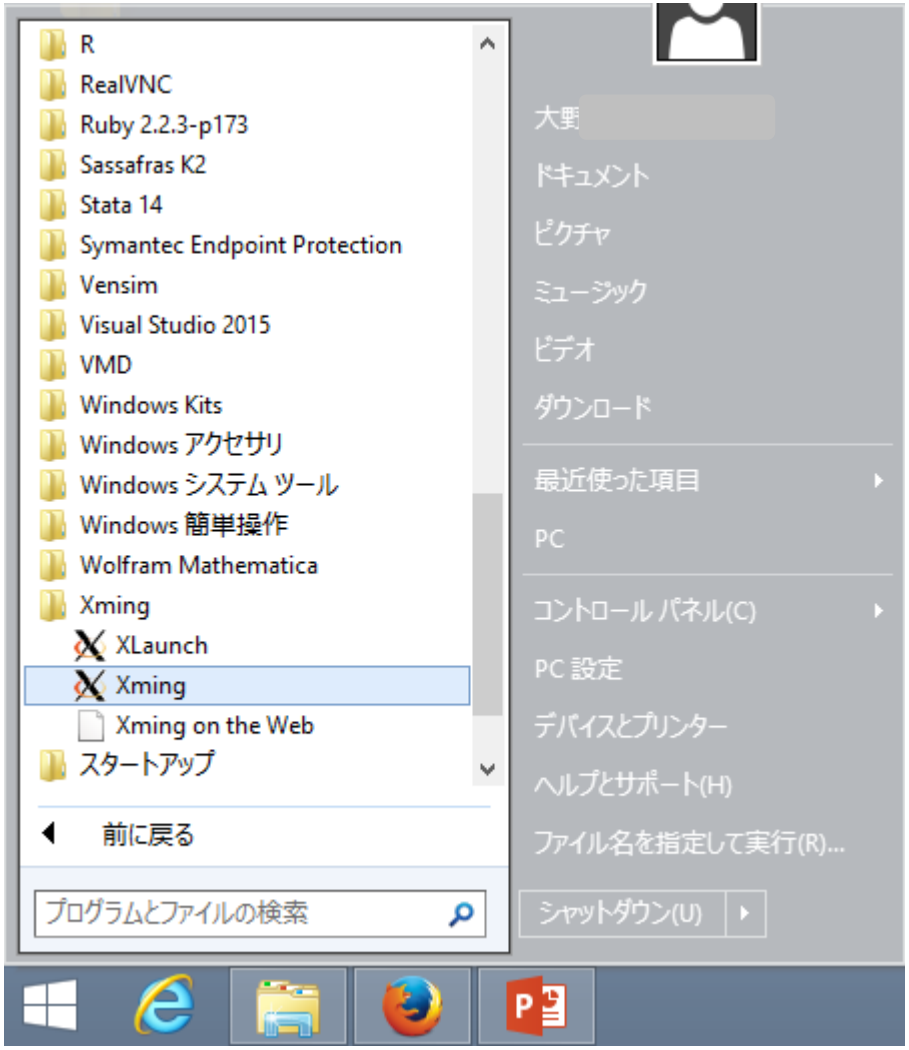
参考

Mike Bailey‘s OpenDX Page: http://web.engr.oregonstate.edu/~mjb/opendx/
D. Thompson, J.Braun, and R.Ford, “*OpenDX: Paths to Visualization*”, Vis Inc., 2001.

OpenDXの起動 -1

X serverが必要

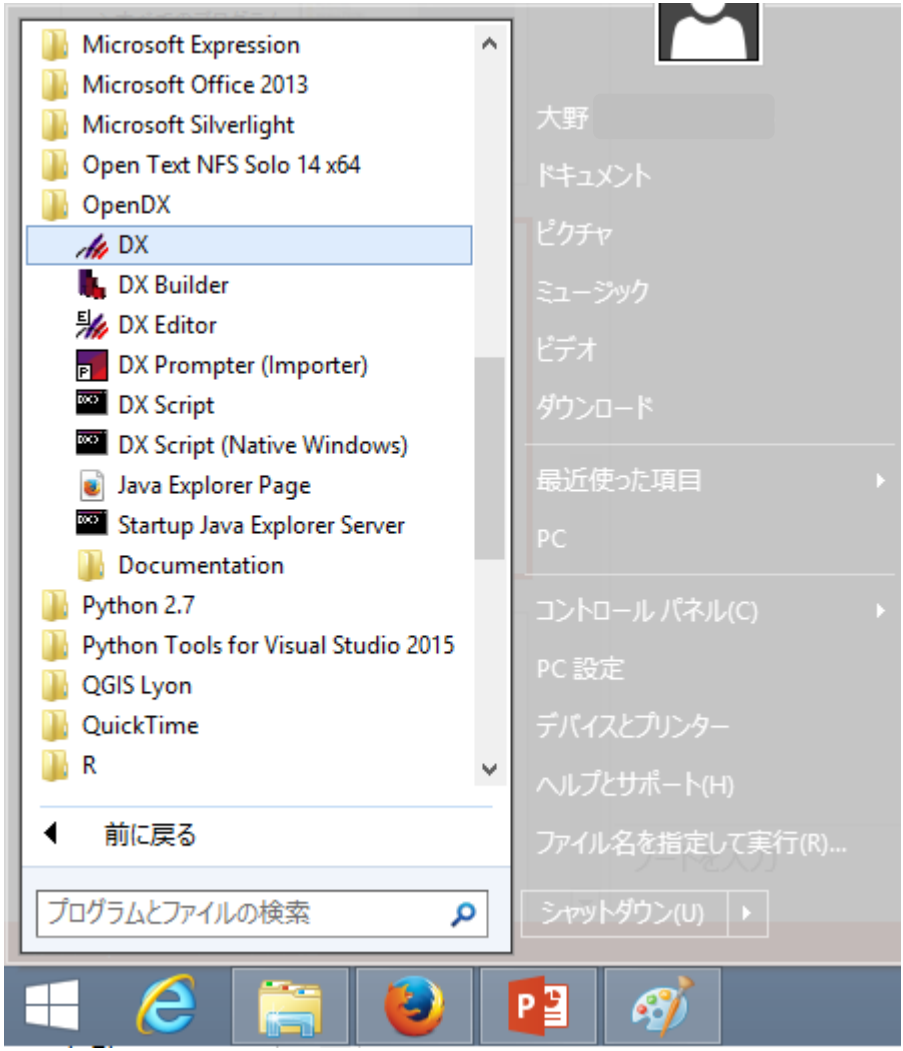
スタートボタン
→すべてのプログラム
→Xming
→Xming
でXサーバを起動



OpenDXの起動 -2

X server起動後

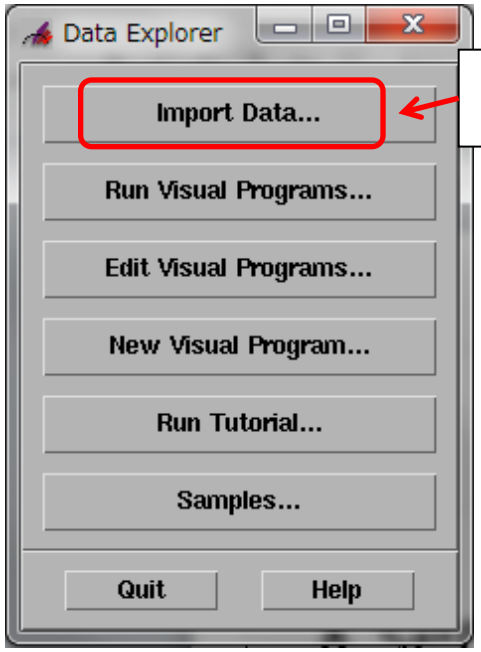
スタートボタン
→すべてのプログラム
→OpenDX
→DX
でOpenDXを起動



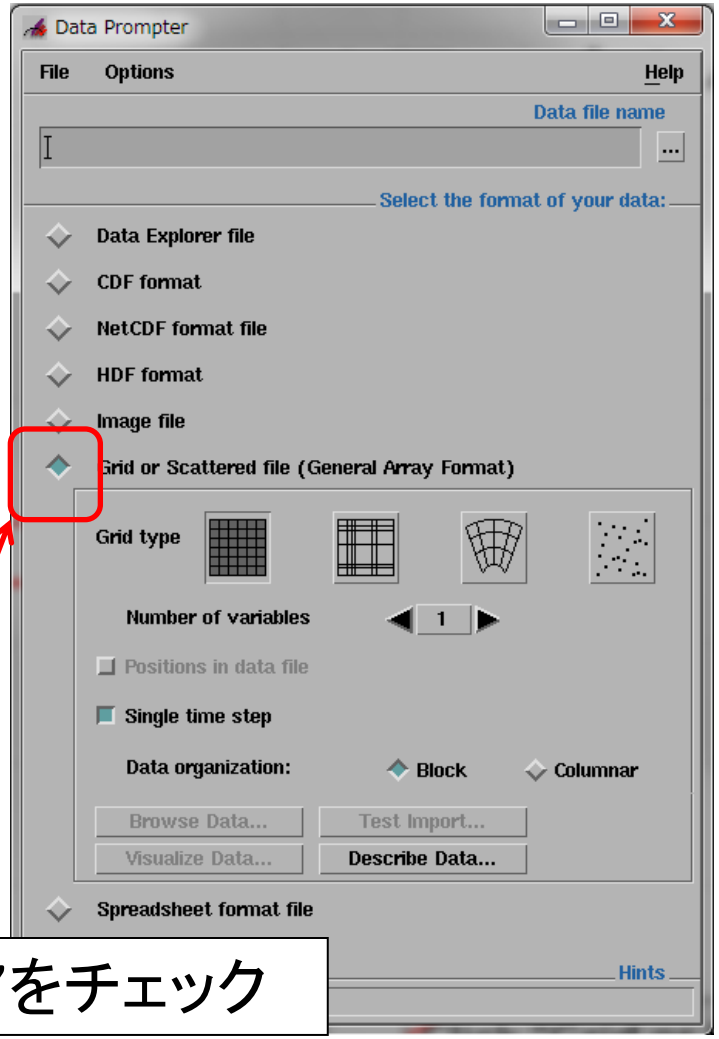
OpenDXの起動 -3

これが出ればOK

General ファイル作成 -1



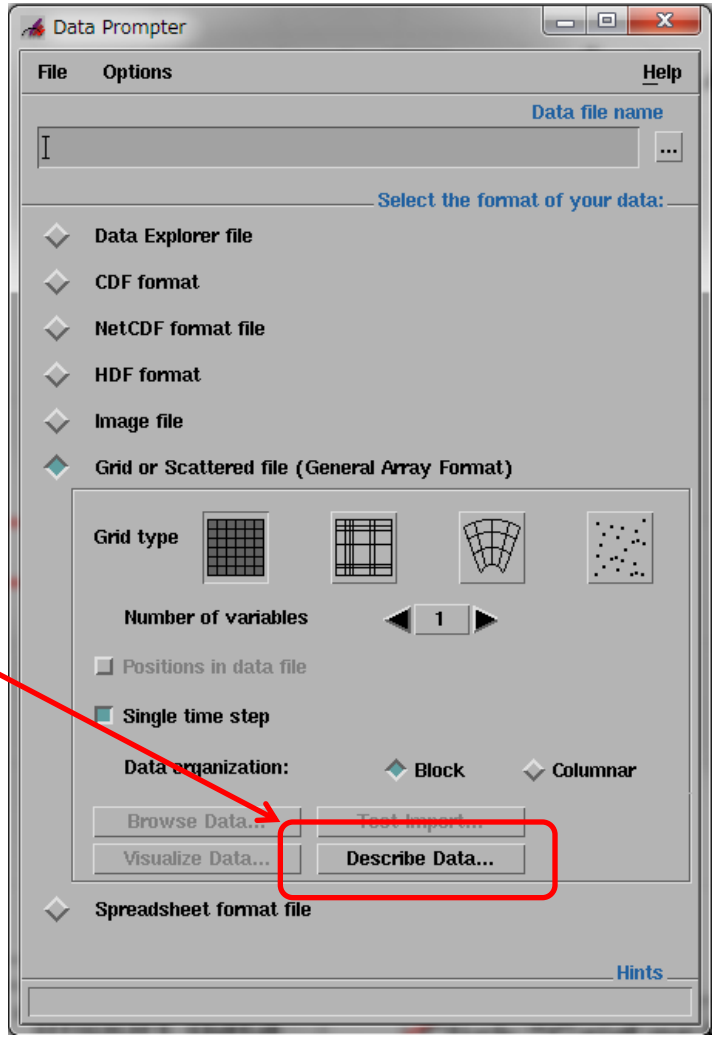
“Import Data”



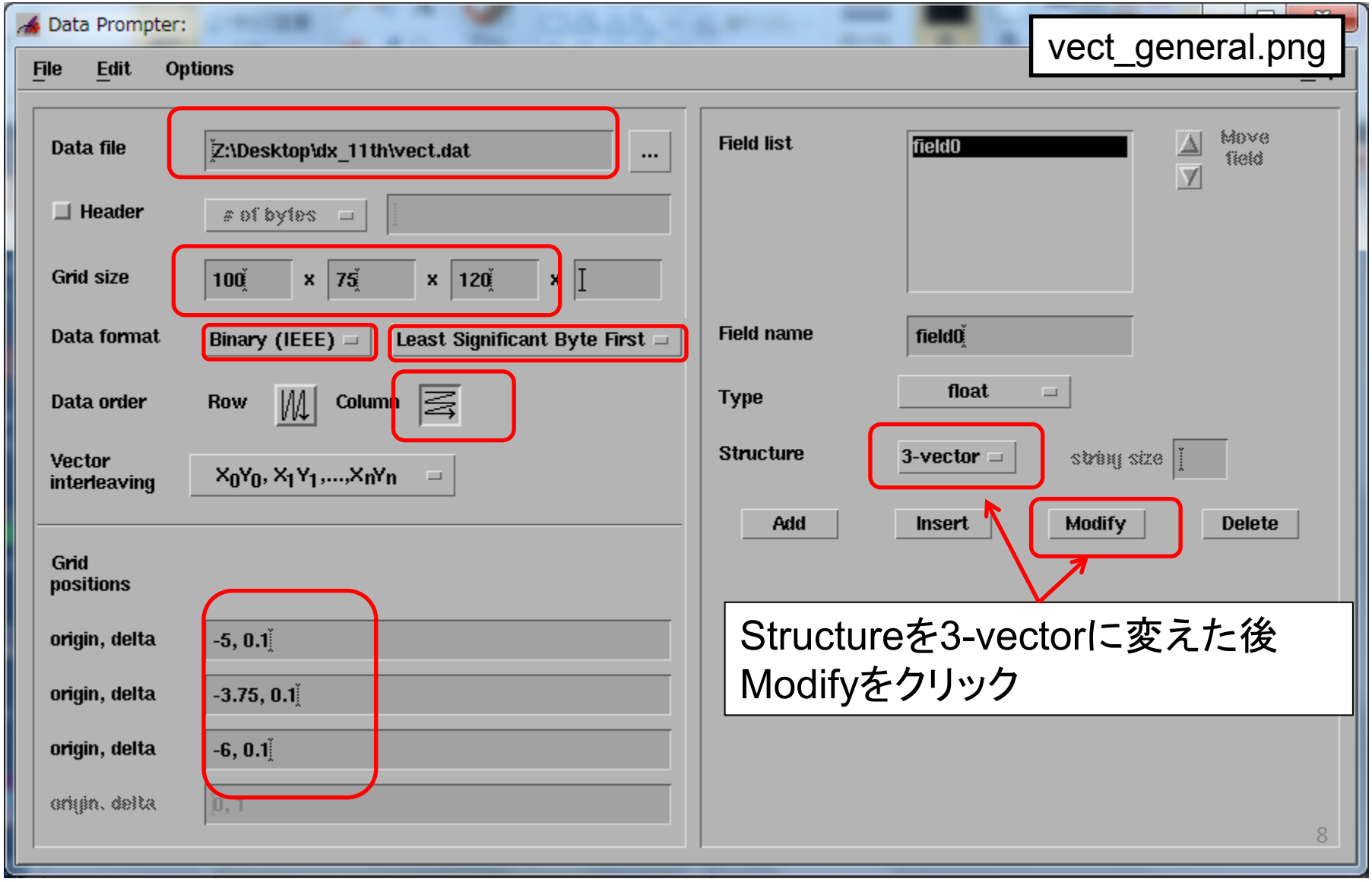
“Grid or Scattered File”をチェック

General ファイル作成 -2

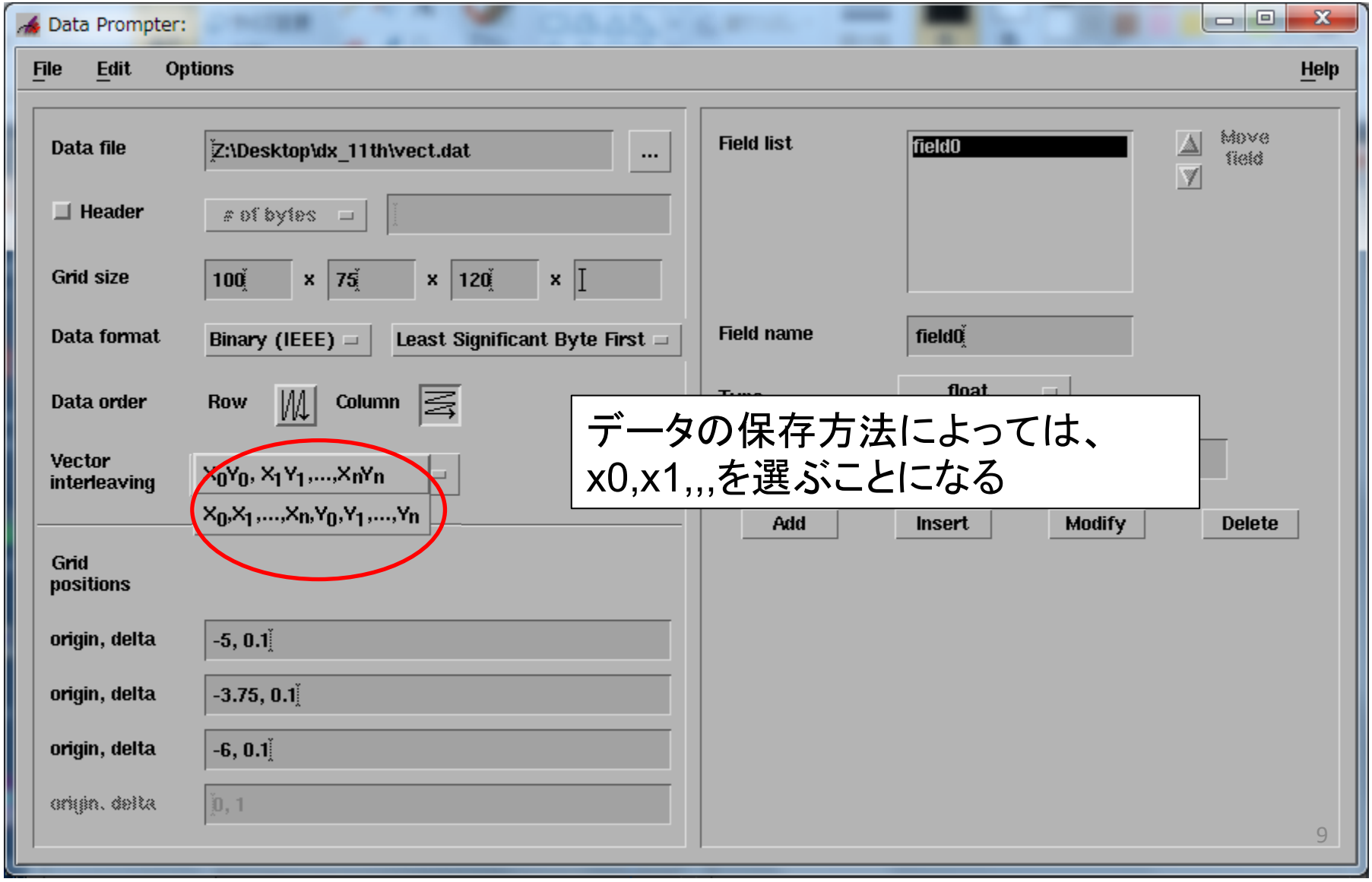
“Describe Data”をクリック



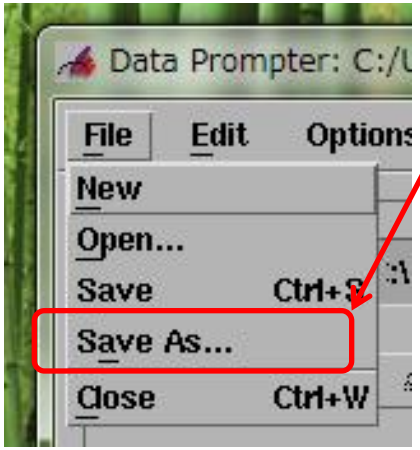
General ファイル作成 -3



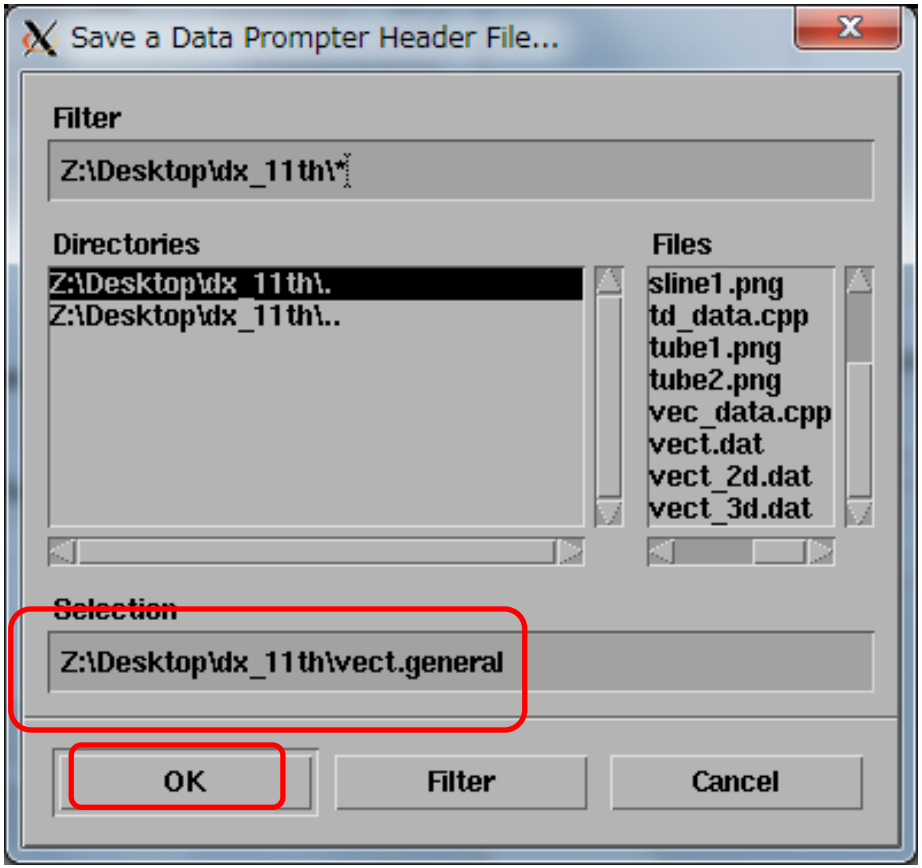
General ファイル作成 -3A



General ファイル作成 -4



File->Save as



vect.generalとして保存

General ファイル作成 -5

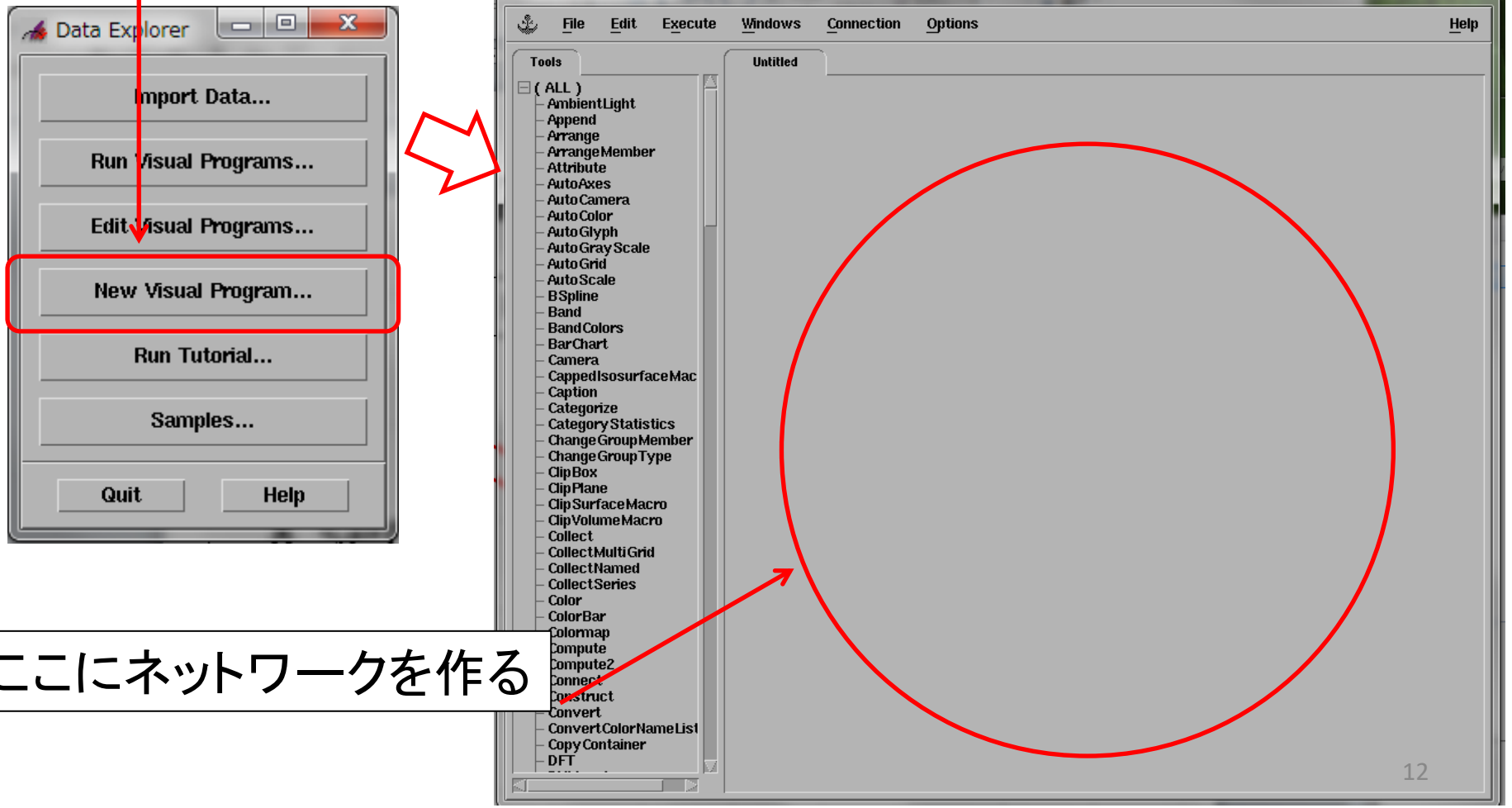
vect.generalの中身

```
file = Z:¥Desktop¥dx_11th¥vect.dat
grid = 100 x 75 x 120
format = lsb ieee
interleaving = record-vector
majority = column
field = field0
structure = 3-vector
type = float
dependency = positions
positions = regular, regular, regular, -5, 0.1, -3.75, 0.1, -6, 0.1

end
```

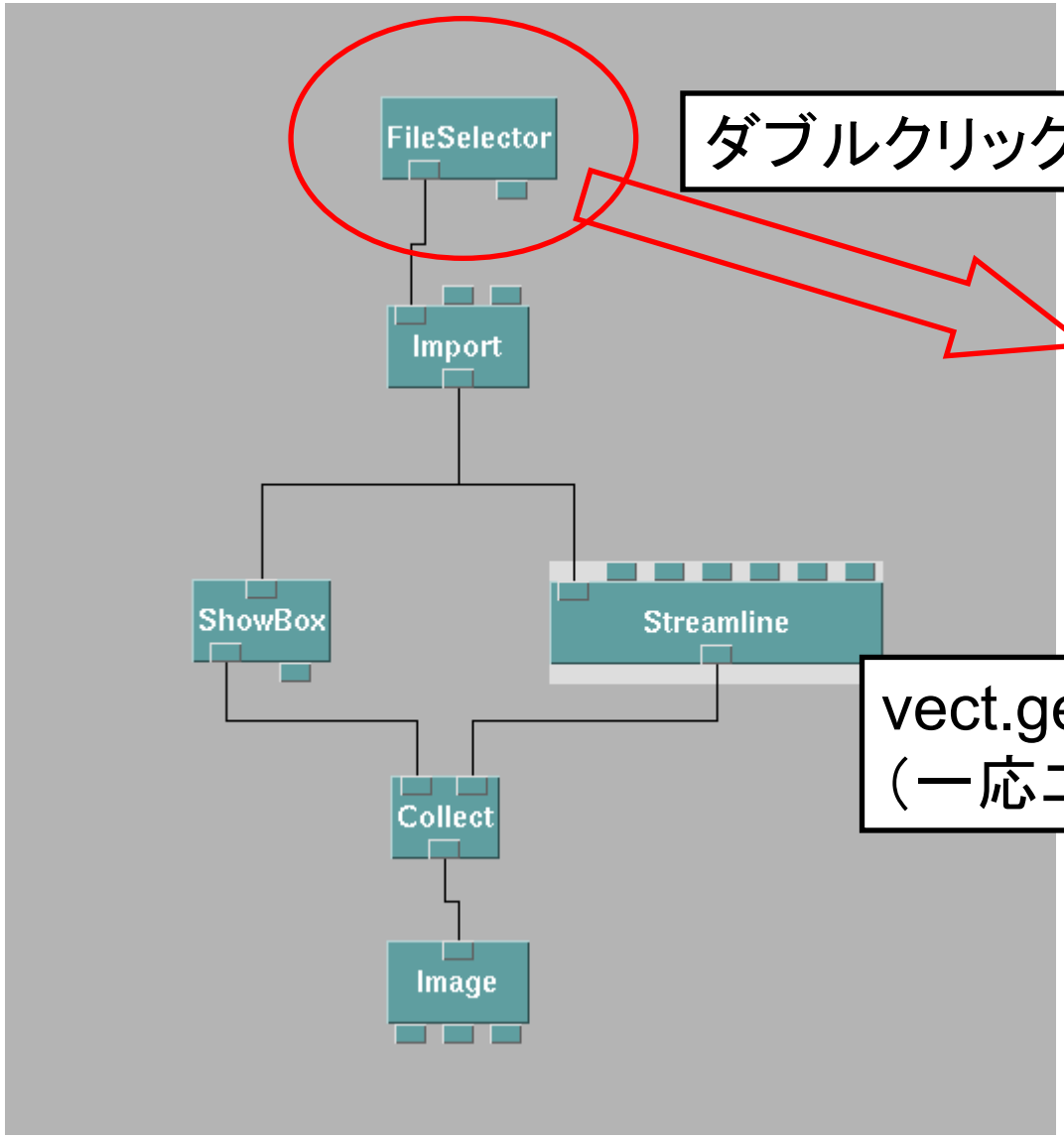
Network作成準備

“New Visual Program”をクリック

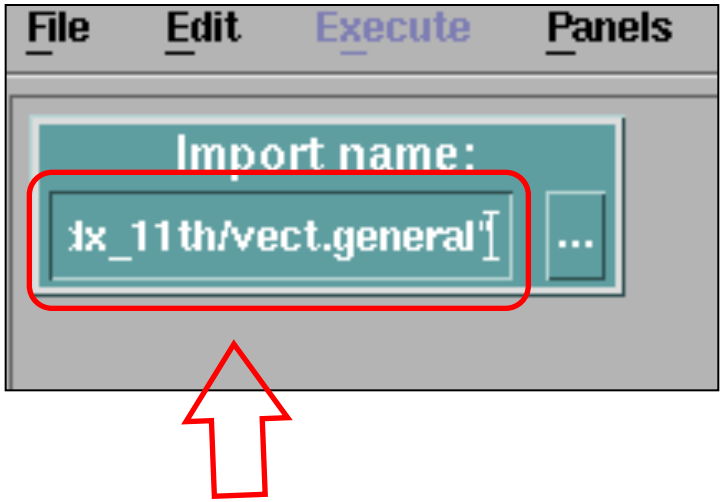


ここにネットワークを作る

StreamLine -1

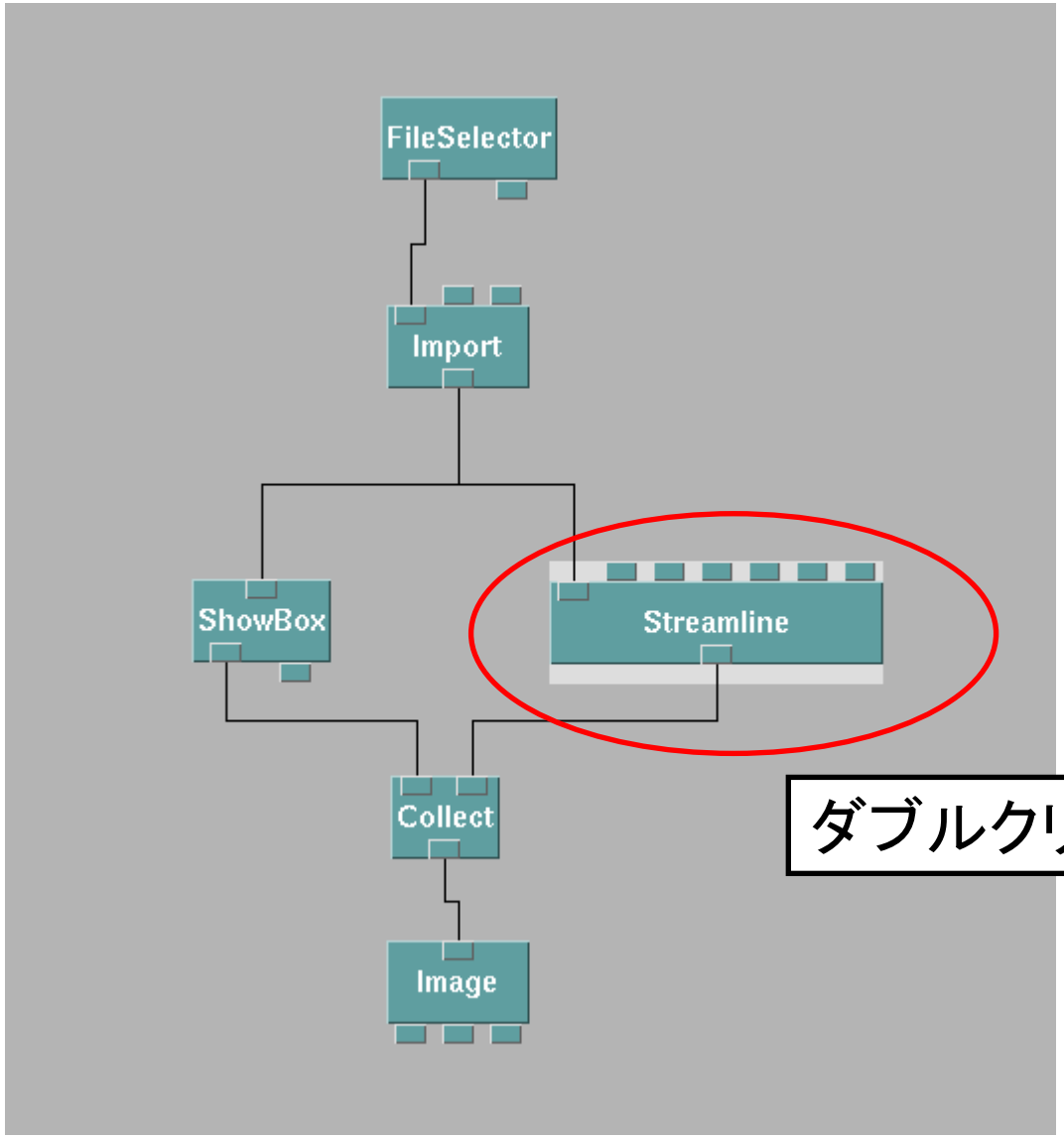


ダブルクリック



vect.generalを入力
（一応エンターキーを押しておく）

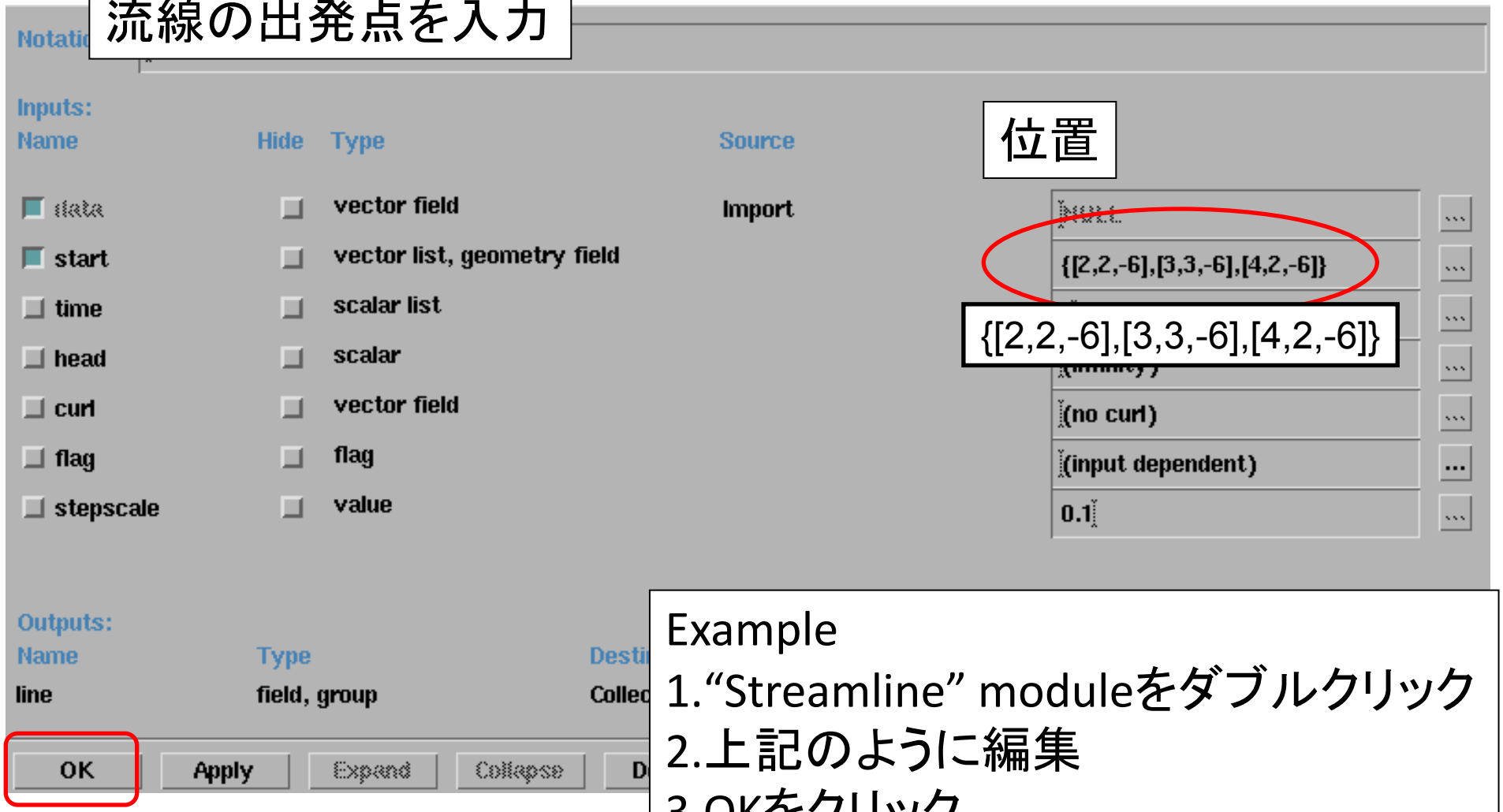
StreamLine -2



流線の出発点を指定する

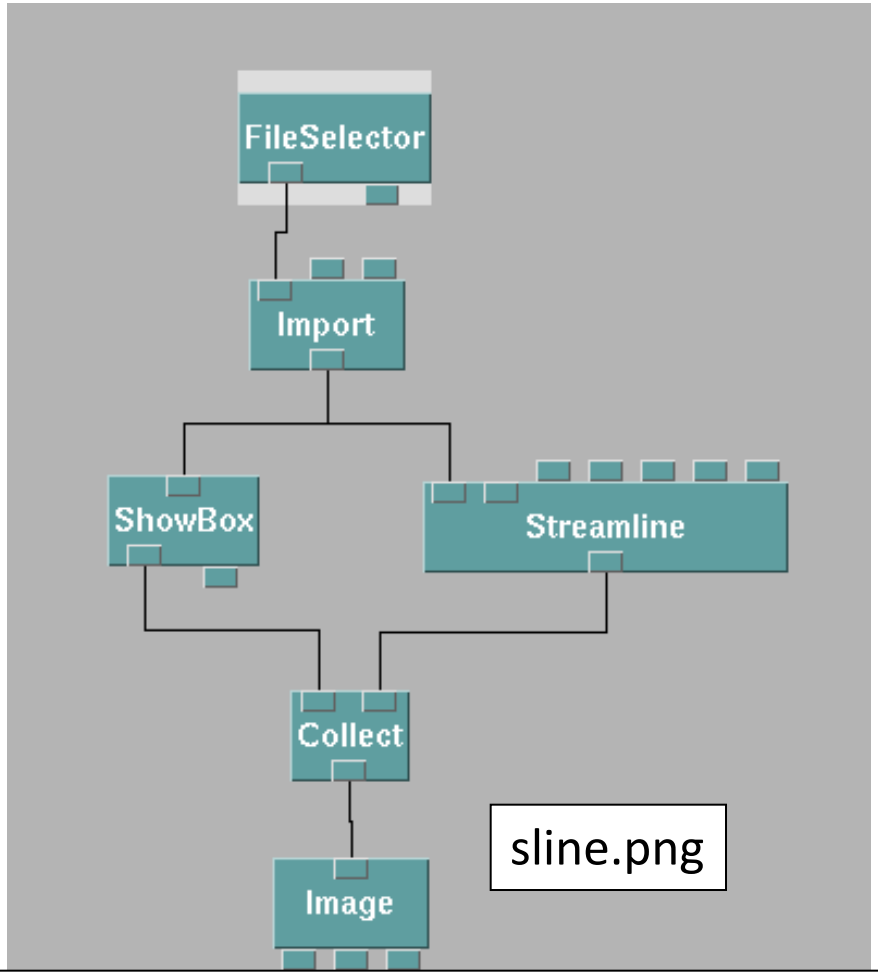
ダブルクリック

StreamLine -3

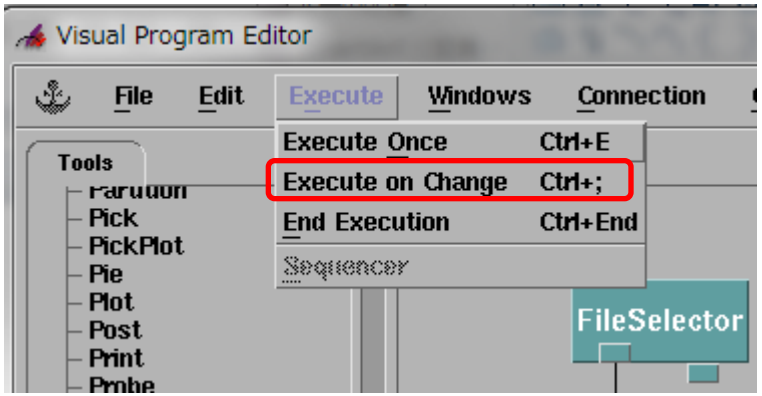


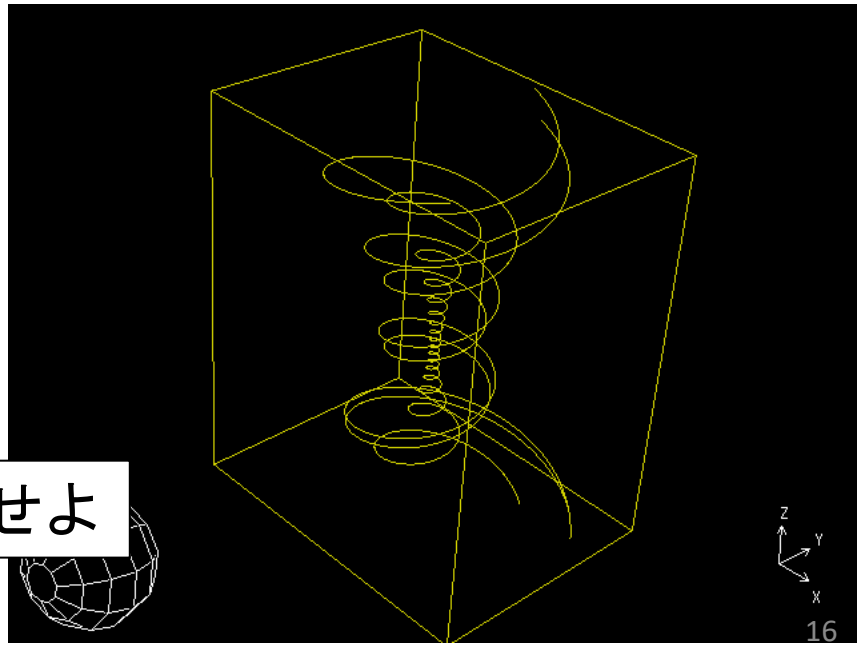
Example
1.“Streamline” moduleをダブルクリック
2.上記のように編集
3.OKをクリック

StreamLine -4



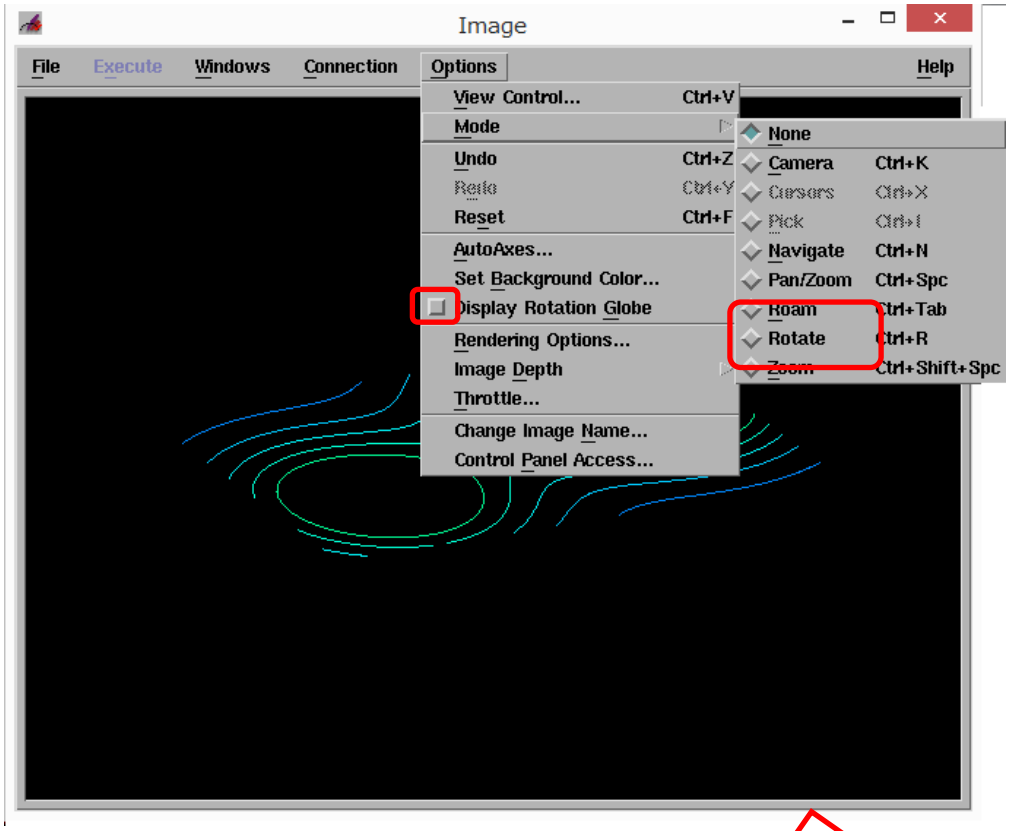
sline.png



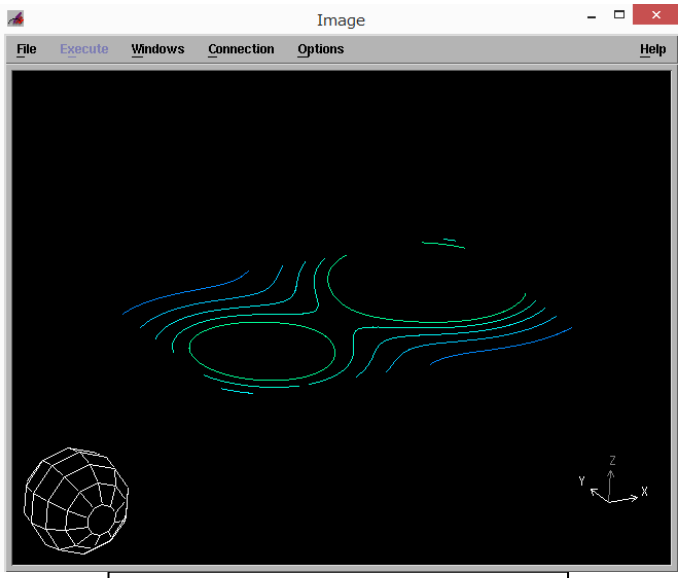


Exercise 1 上記ネットワークを作成せよ

Windowの操作:再掲



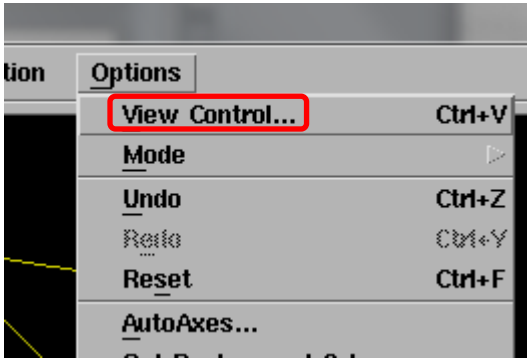
1. Option
2. Mode
3. “Rotate”をチェック
4. (“Display Rotation Globe”もチェック)

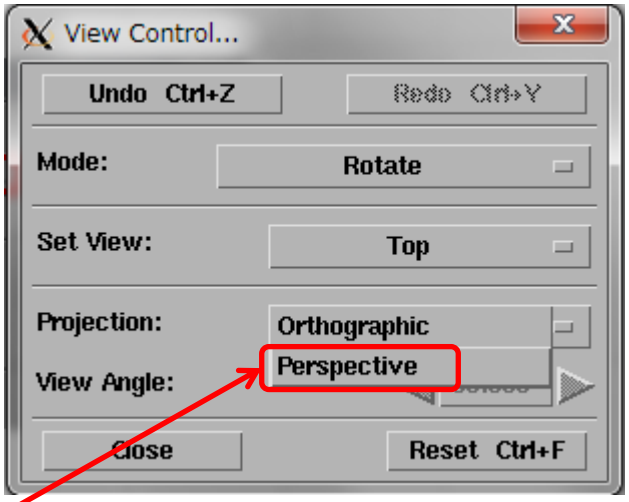


Zoom も試してみるとよい

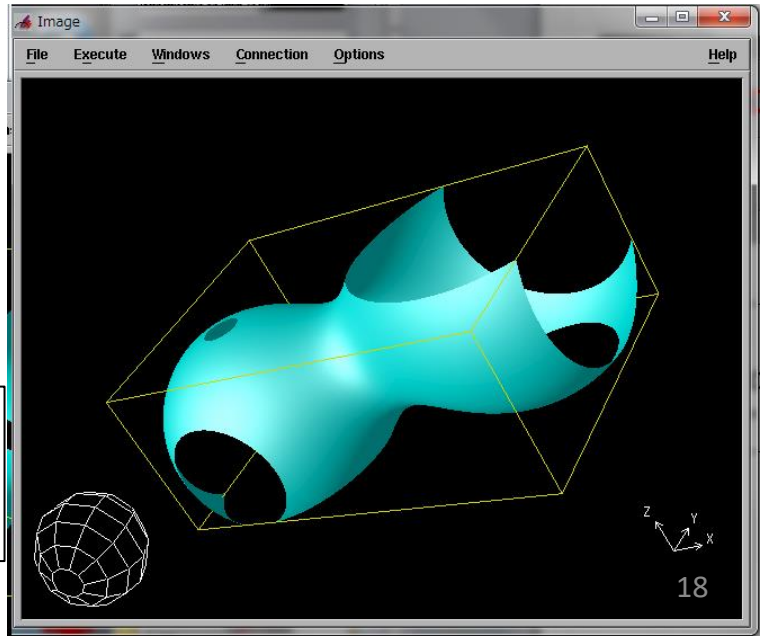
マウスで回せる

Windowの操作：再掲





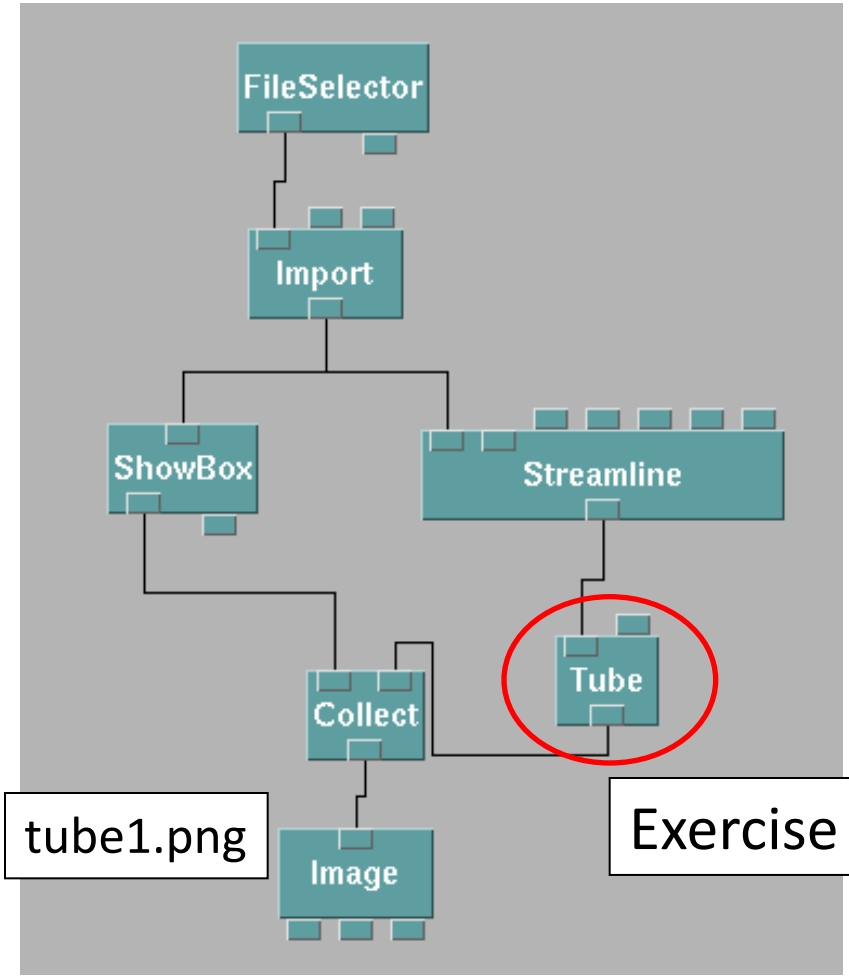
1. Option
2. View Control
3. “Perspective”

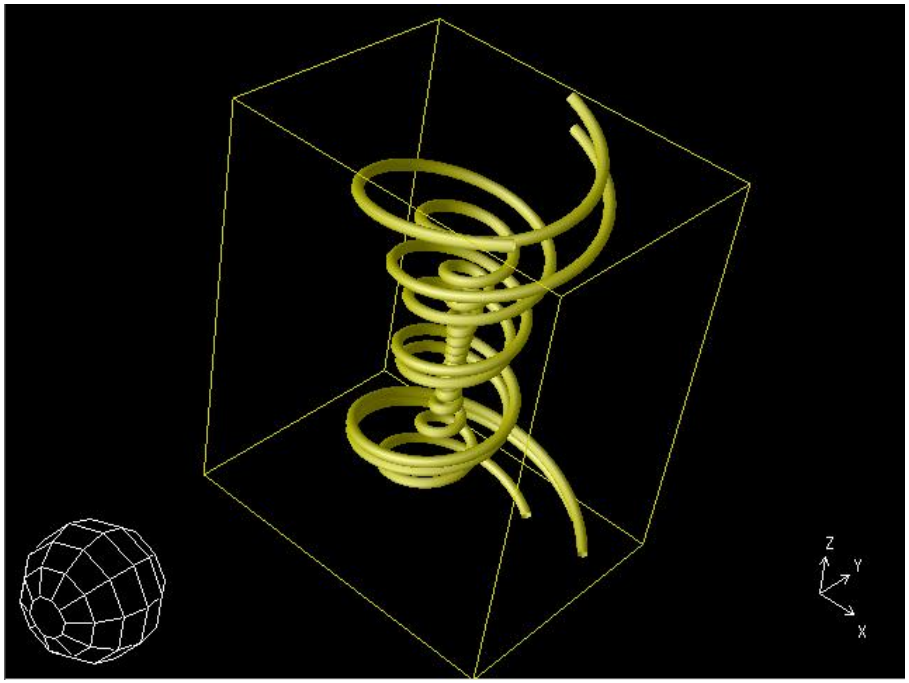


透視射影になる
Ortho->Perspective

Tube -1

流線をTubeにして迫力を出す



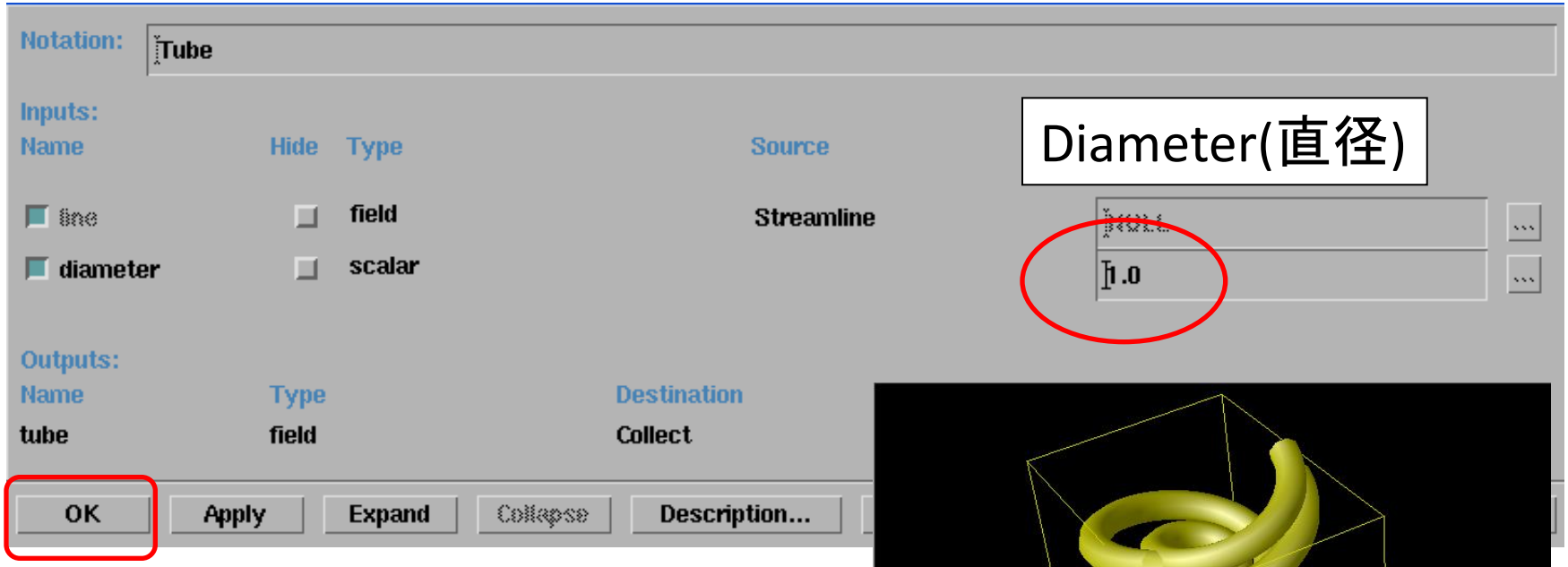


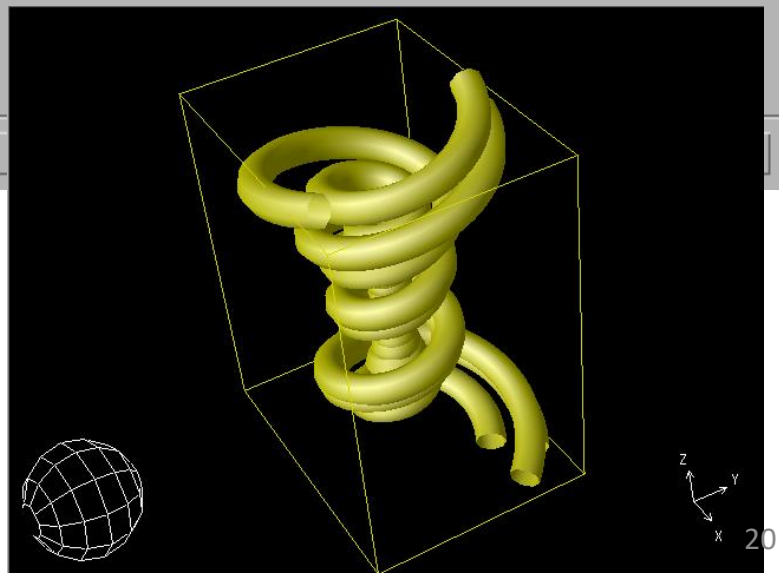
tube1.png

Exercise 2 上記ネットワークを作成せよ

Tube -2

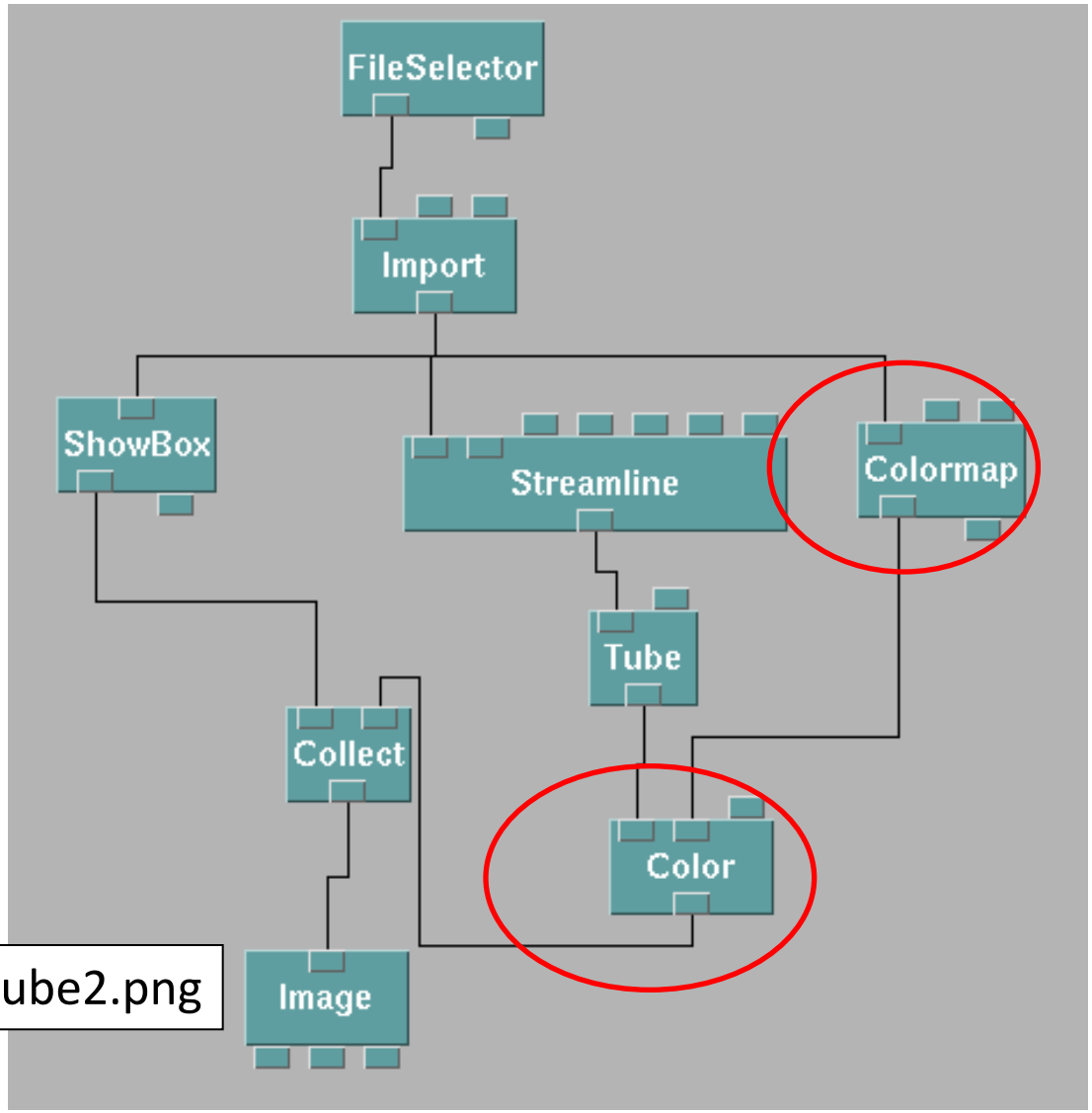
チューブの直径を変えられる





Example
1.“Tube” moduleをダブルクリック
2.上記のように編集
3.OKをクリック

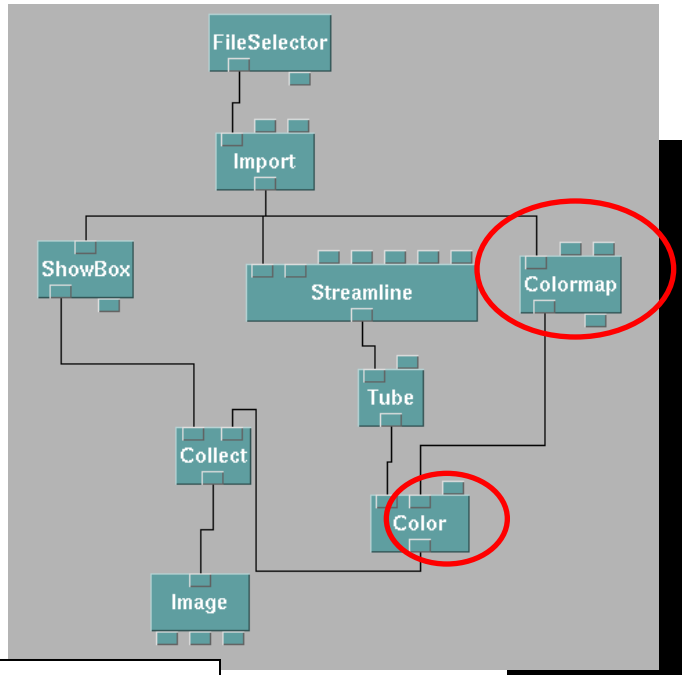
Tube -3



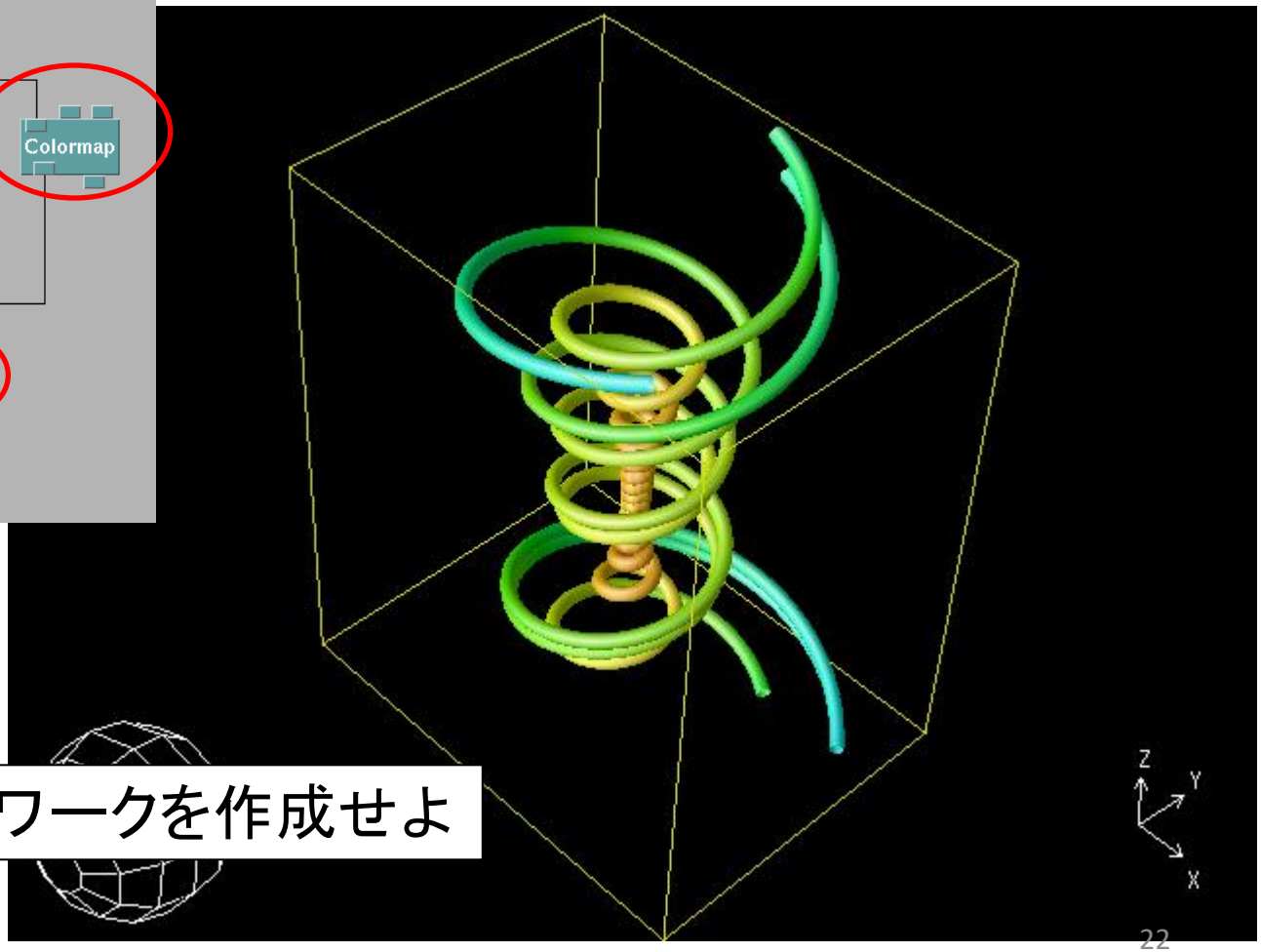
tube2.png

Tubeにベクトル場の大きさに応じて色付けする(流線でもできる)

Tube -4



tube2.png



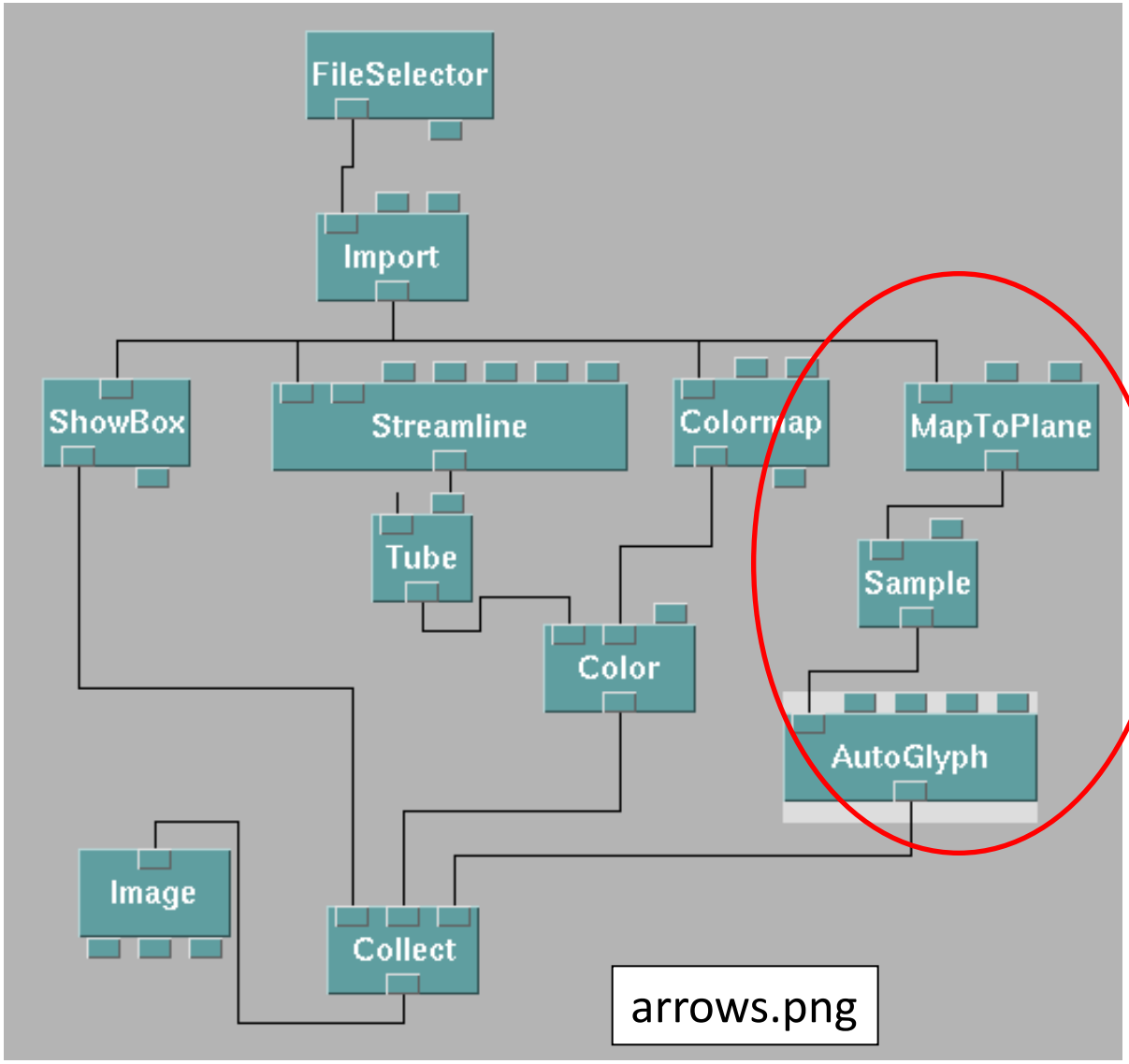
Exercise 3 上記ネットワークを作成せよ

Arrows -1

ベクトル場を矢印で表現する;

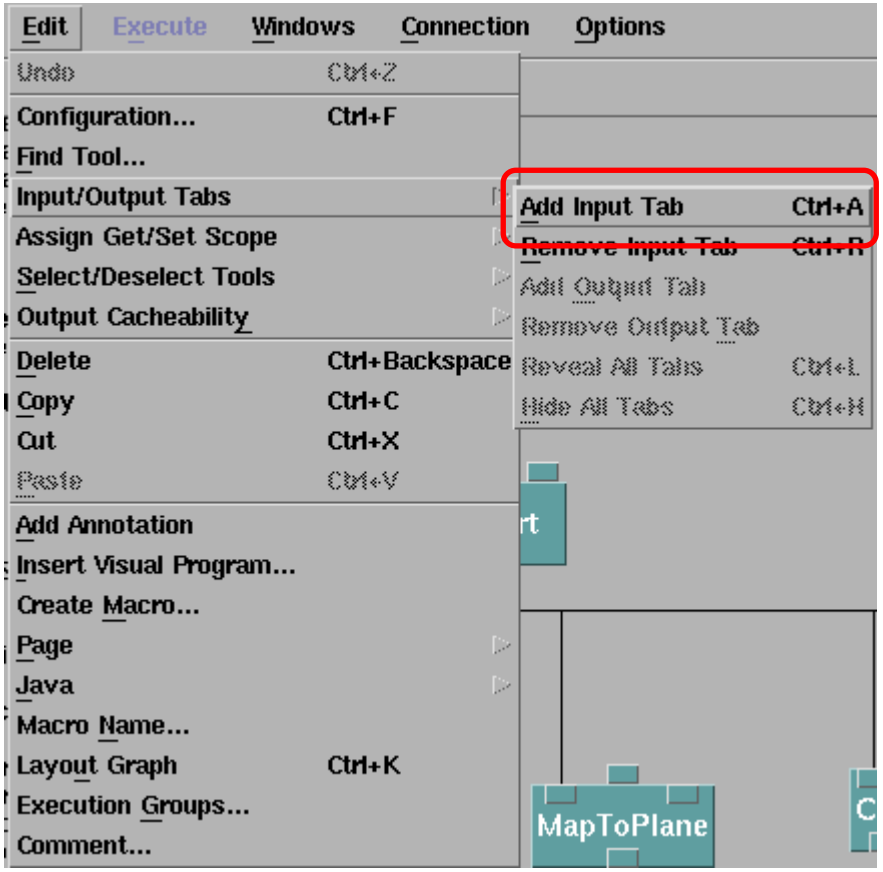
ただし、データ全体に矢印を表示すると、かえってわかりにくいので、一部分にのみ矢印を配置する

Arrows -2

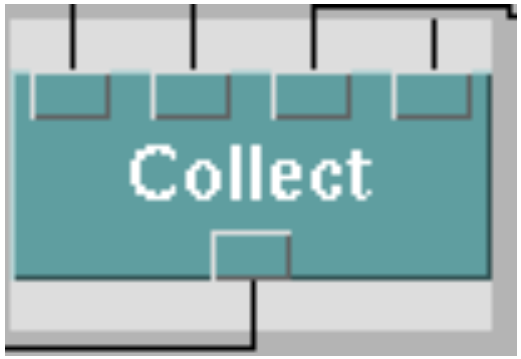

FileSelector
Import
ShowBox
Streamline
Colormap
MapToPlane
Tube
Sample
Color
AutoGlyph
Image
Collect
arrows.png

第10回Appendix -1

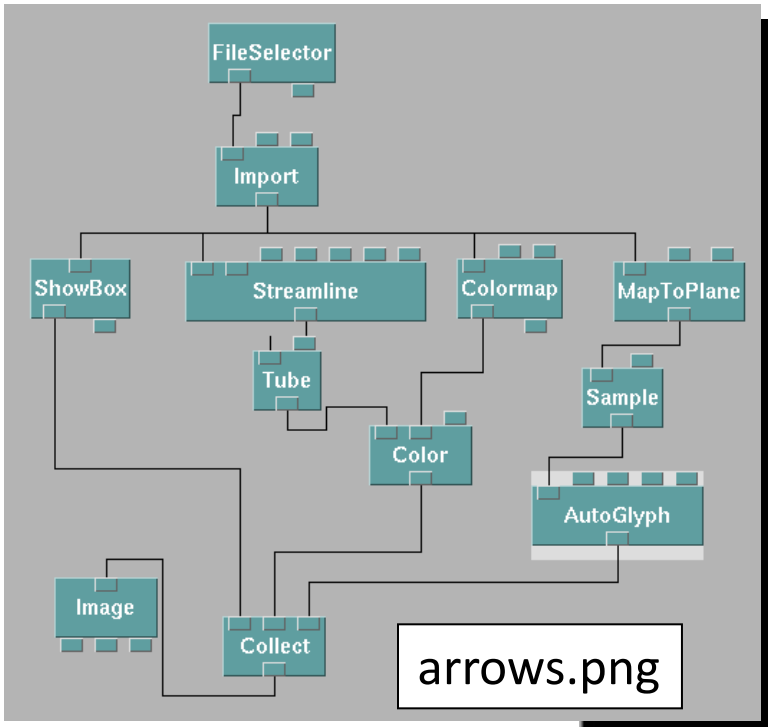
- Collectの入り口を増やす



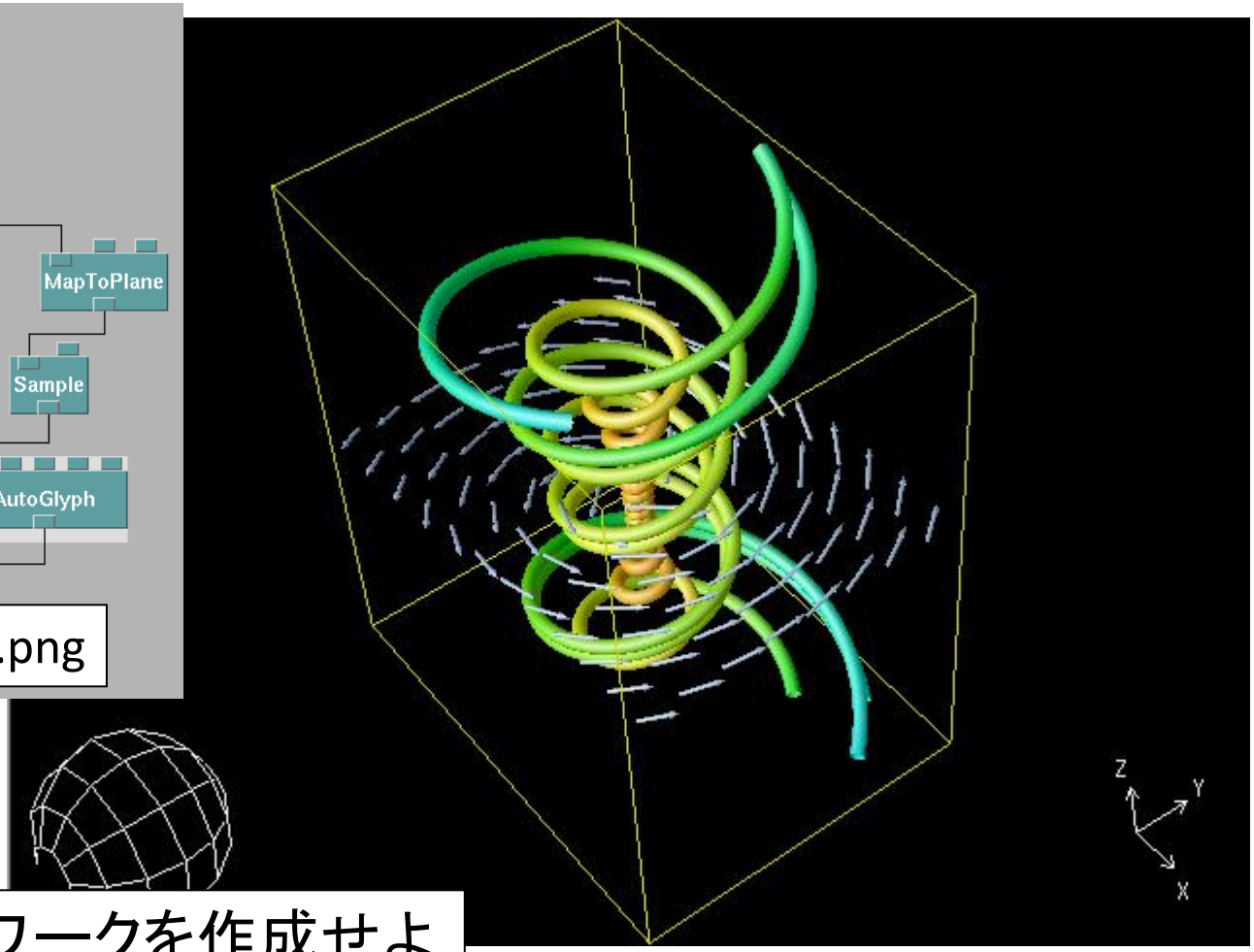
1. “Collect” moduleをクリック
2. Edit -> “Input/Output Tabs” -> “Add Input Tabs”



Arrows -3



arrows.png



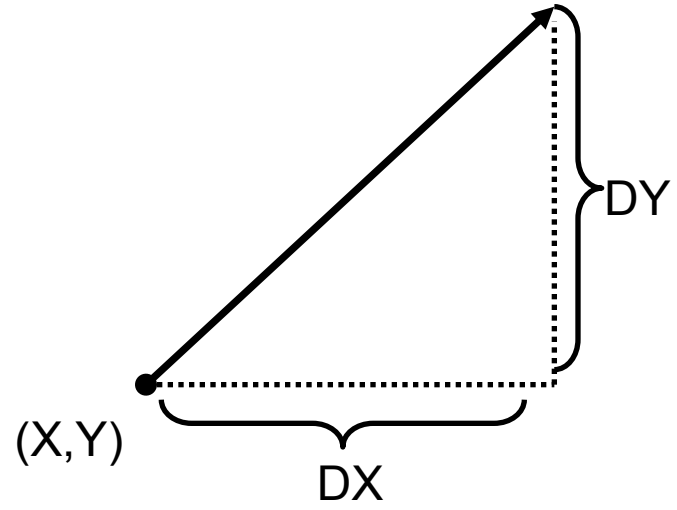
Exercise 4 上記ネットワークを作成せよ

Appendix A-1

- Gnuplotによるベクトル場の可視化
- 2次元ベクトル場

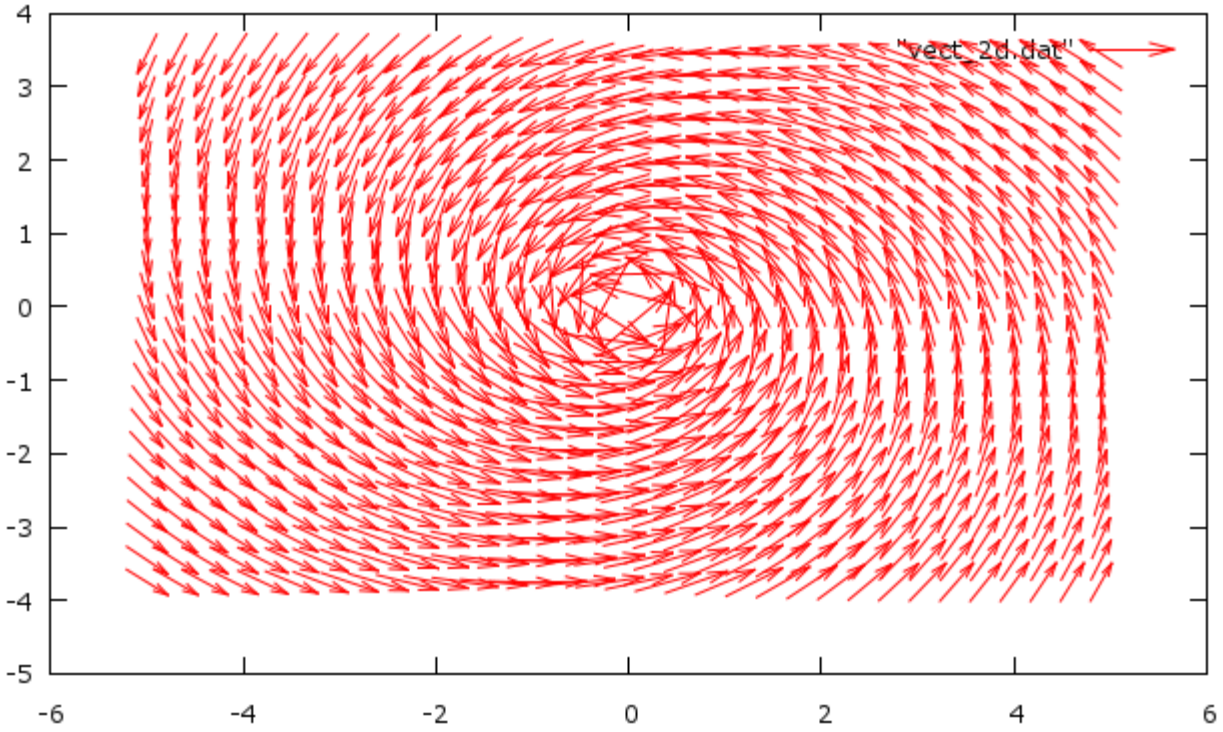
データフォーマット

X1 ␣ Y1 ␣ DX1 ␣ DY1⏎
X2 ␣ Y2 ␣ DX2 ␣ DY2⏎
…



Appendix A-2

gnuplot> plot␣“vect_2d.dat”␣with␣vectors⏎



矢印の長さを短くしたほうがよさそう

Appendix A-3

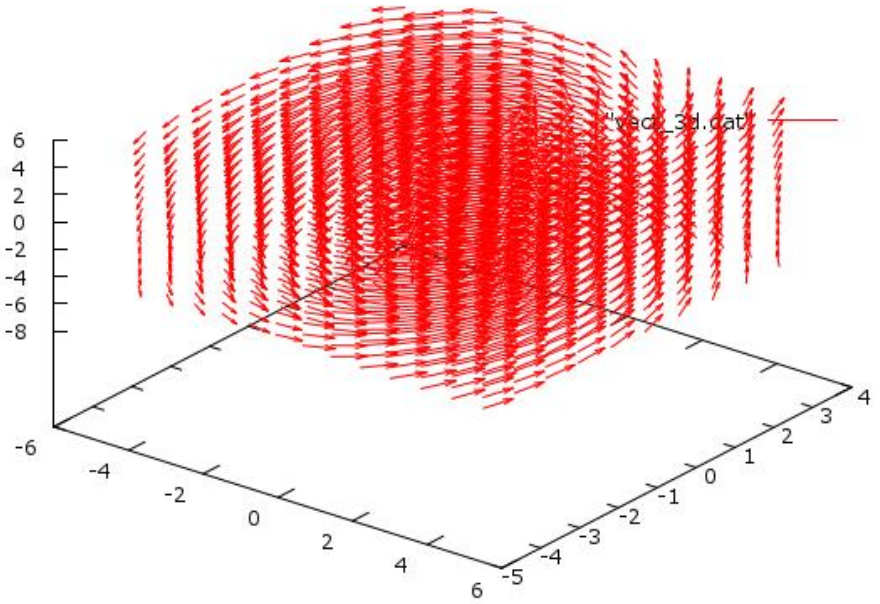
3次元ベクトル場

データフォーマット

X1 ␣ Y1 ␣ Z1 ␣ DX1 ␣ DY1 ␣ DZ1⏎

X2 ␣ Y2 ␣ Z2 ␣ DX2 ␣ DY2 ␣ DZ2⏎

…



gnuplot> splot ␣ “vect_3d.dat” ␣ with ␣ vectors⏎

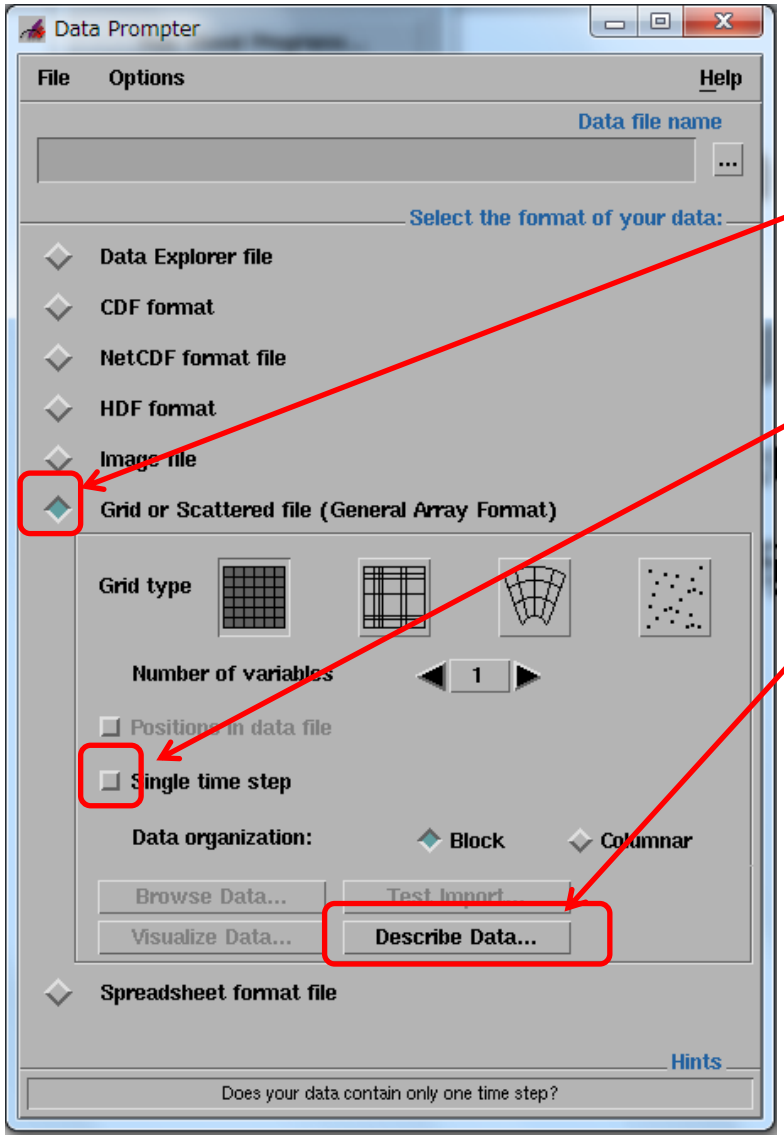
vect_2d.dat, vect_3d.datは、gp_data.cppで作成した。
いずれも矢印の長さを短くしたほうがよさそう。
gp_data.cppをどう変えればよいか考えてみよう。

Appendix B-1

- OpenDXによる時系列データの取り扱い
- scal.datを使う(80x70x60, T=1～20)

1. .generalファイル作成法
2. サンプルネットワーク
3. 連番画像
   - 連番画像保存
   - 連番画像処理

Appendix B-2



1. Grid or Scattered fileをクリック
2. “Single time step”を解除
3. Describe Dataをクリック

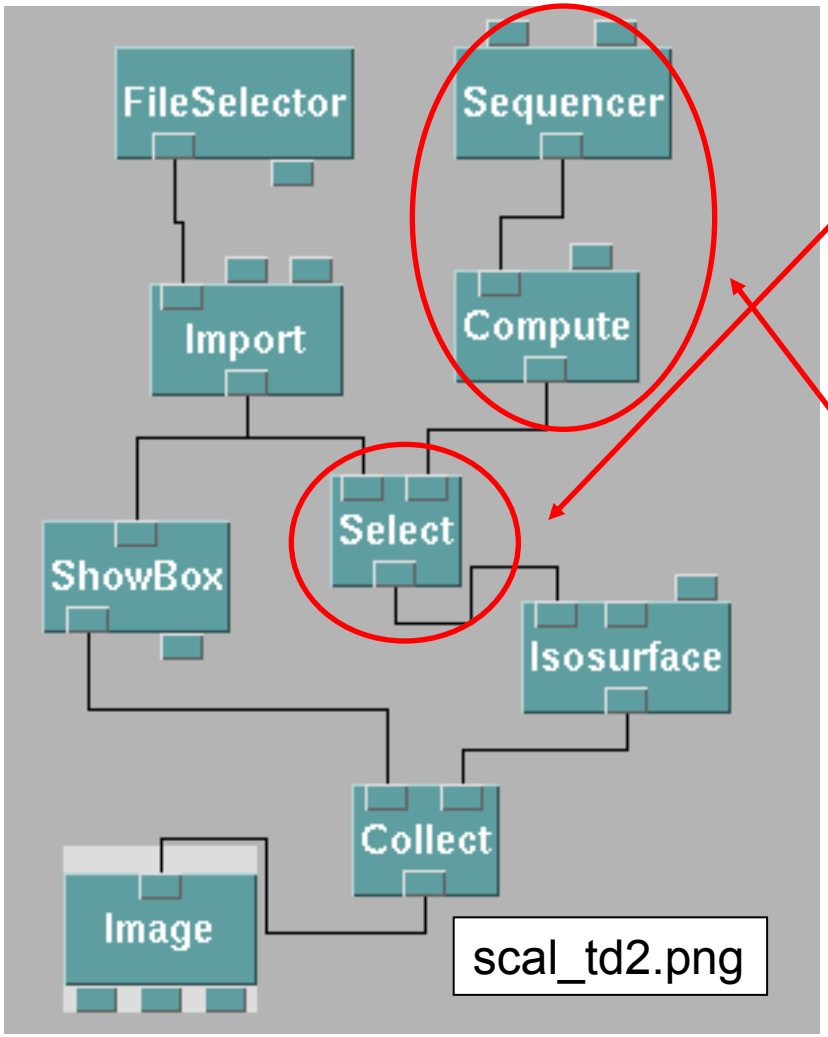
Appendix B-3

時刻数を入れる

scal_td1.png

File -> Save asで
scal_td.generalとして保存

Appendix B-4

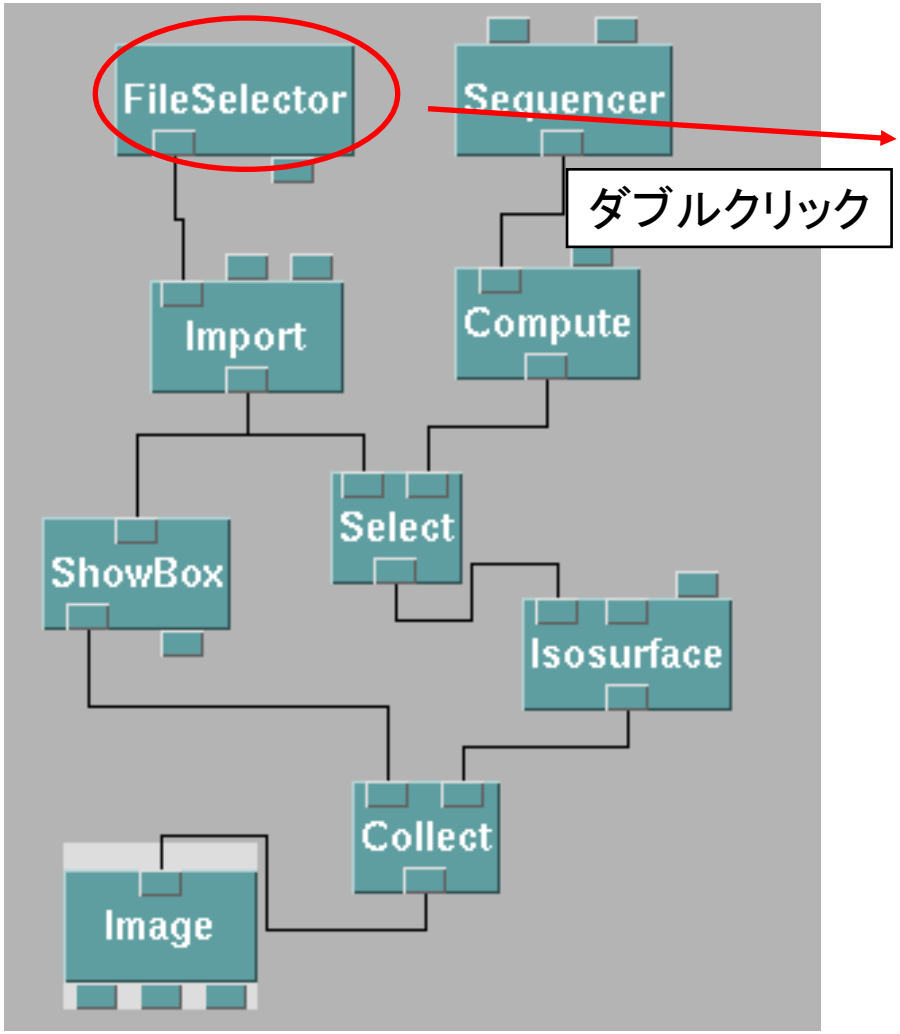


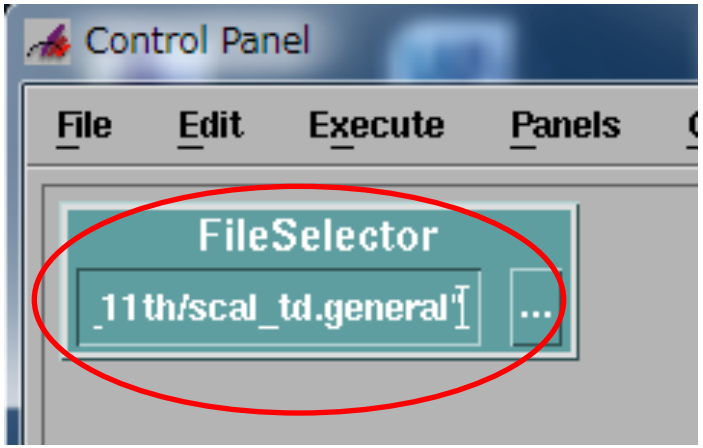
scal_td2.png

Selectで時刻を選ぶ

SequencerとComputeでインタラクティブに時刻を選ぶ

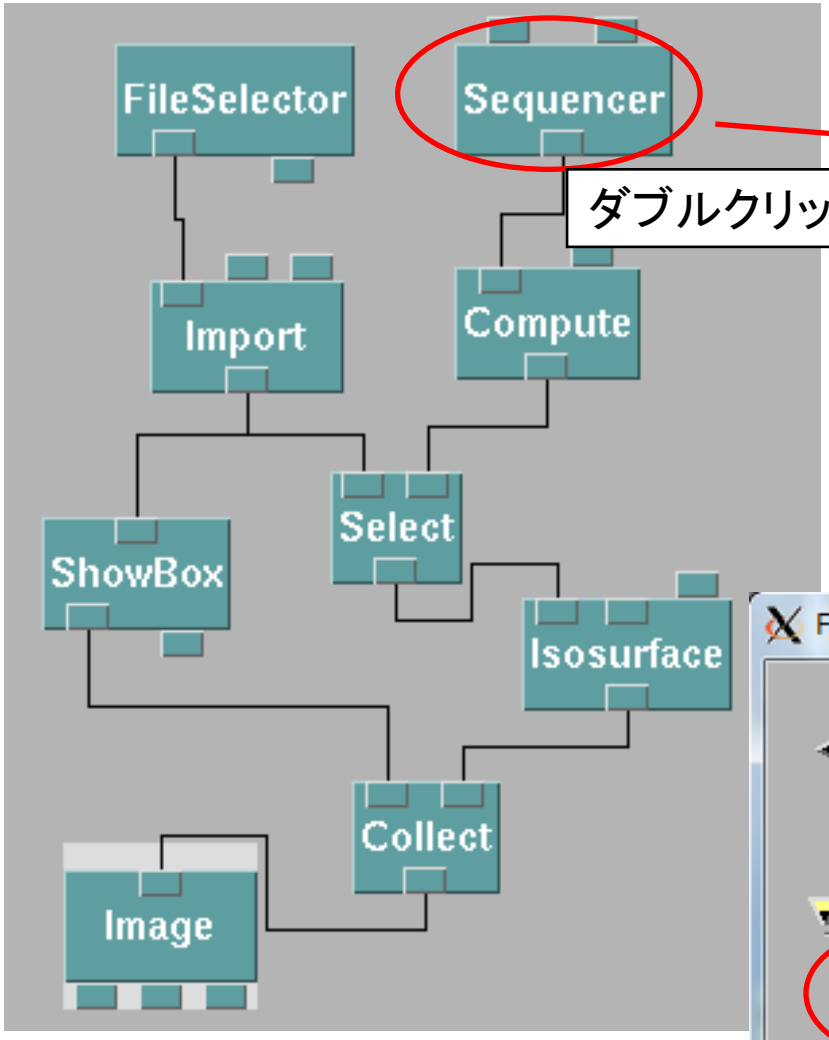
Appendix B-5





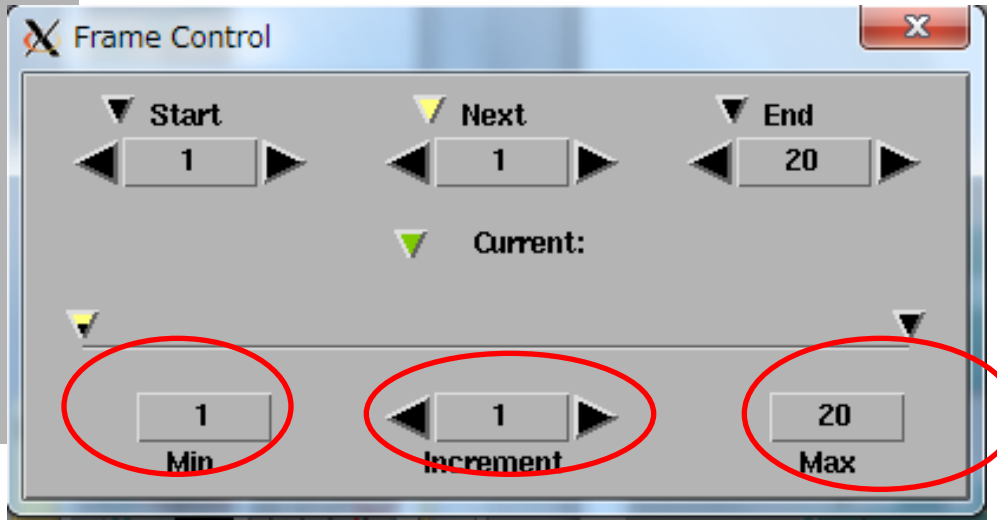
scal_dt.generalをインプット

Appendix B-6

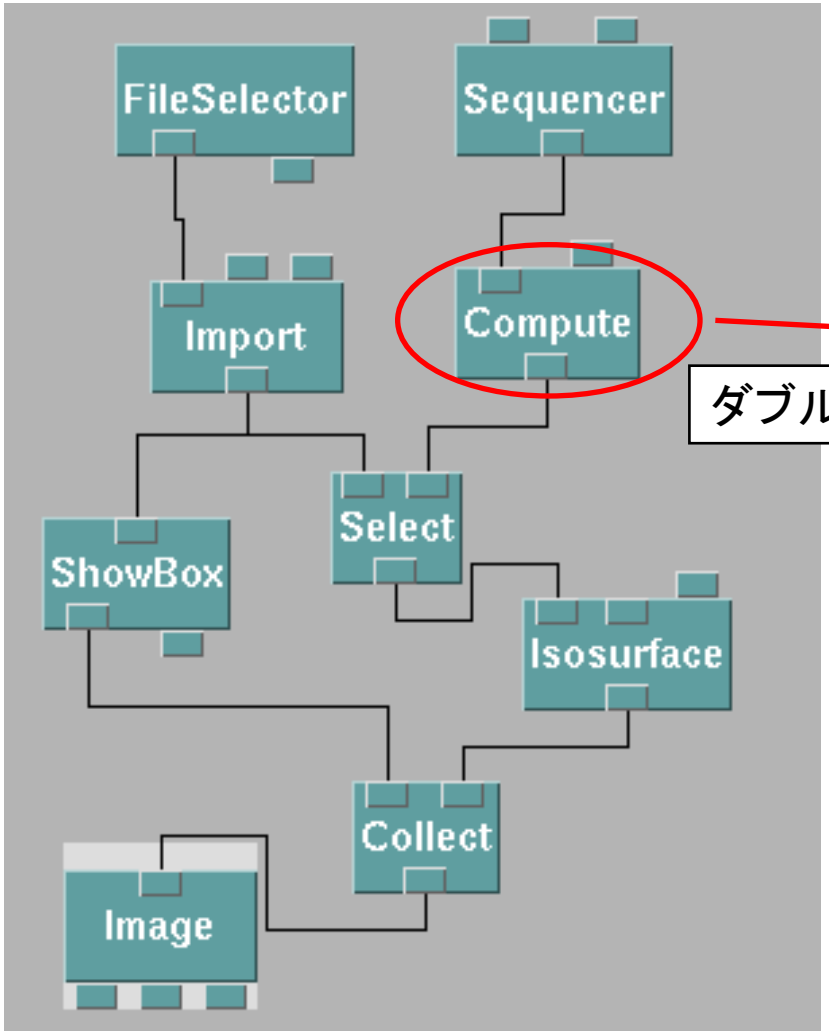


ダブルクリック

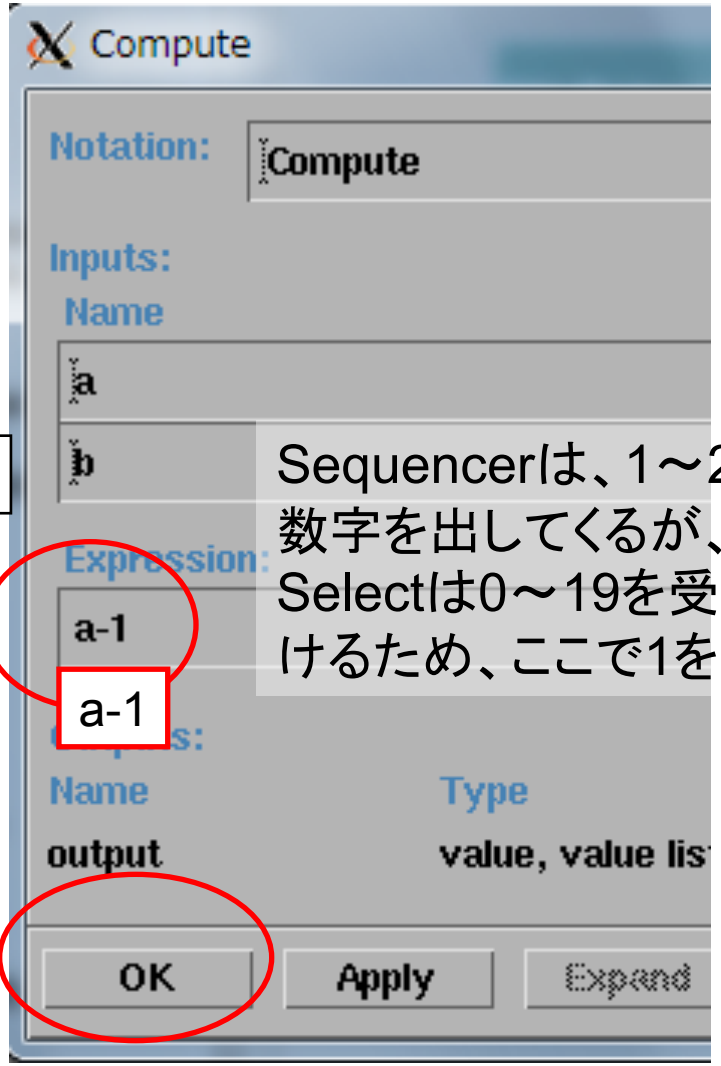
クリック



Appendix B-7



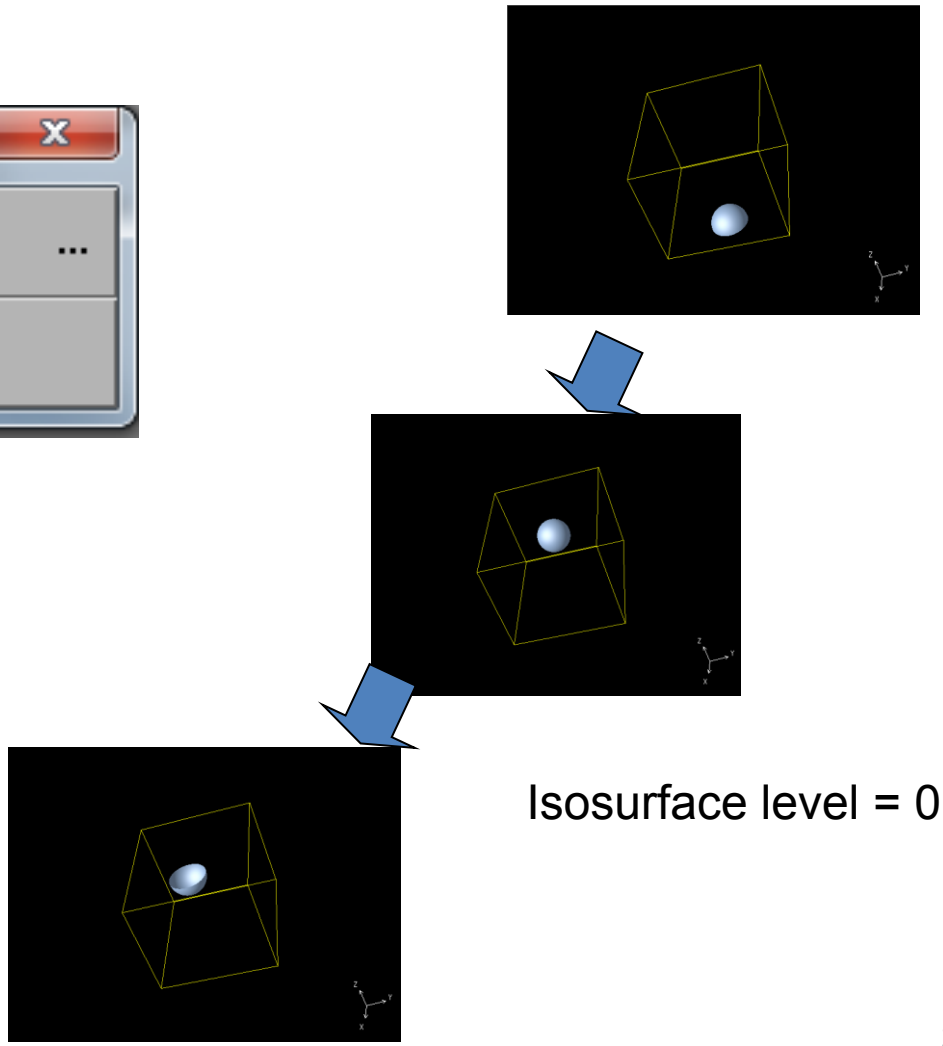
ダブルクリック



Sequencerは、1～20の数字を出してくるが、Selectは0～19を受け付けるため、ここで1を引く

Appendix B-8

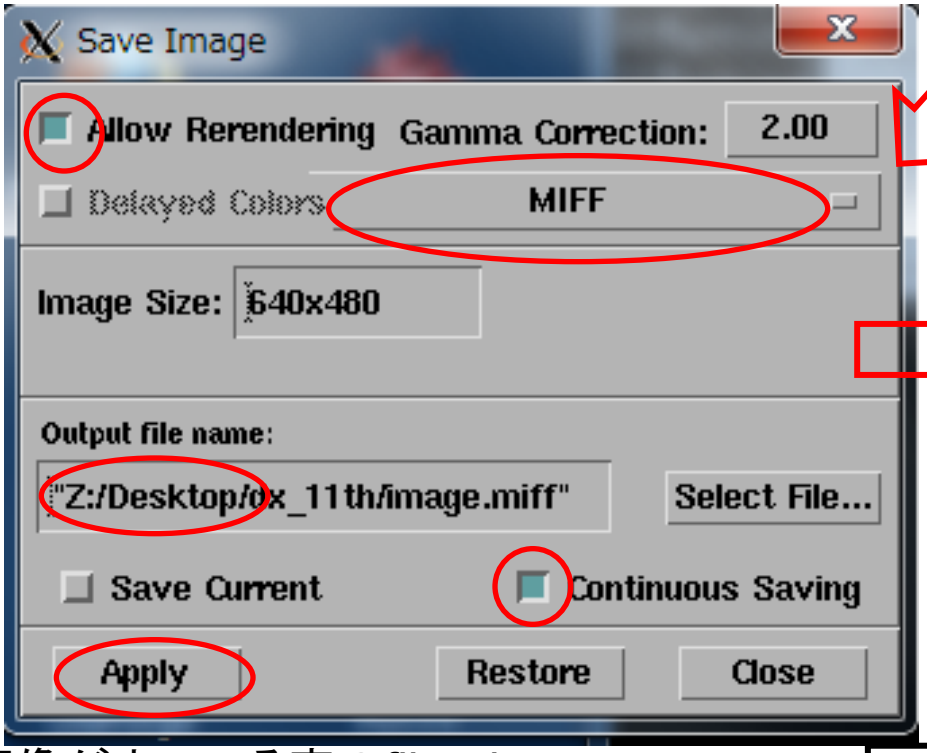
このボタンで再生できる



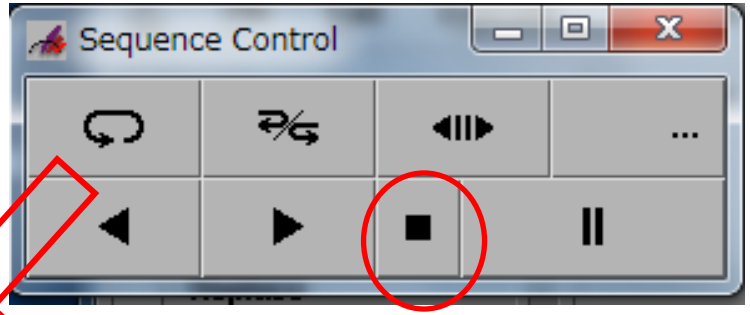
Isosurface level = 0.9

Appendix B-9

画像保存法



画像が出ている窓のfile->image save



再生が終了したら
- Continuous Saving をoff
- Close

Appendix B-10

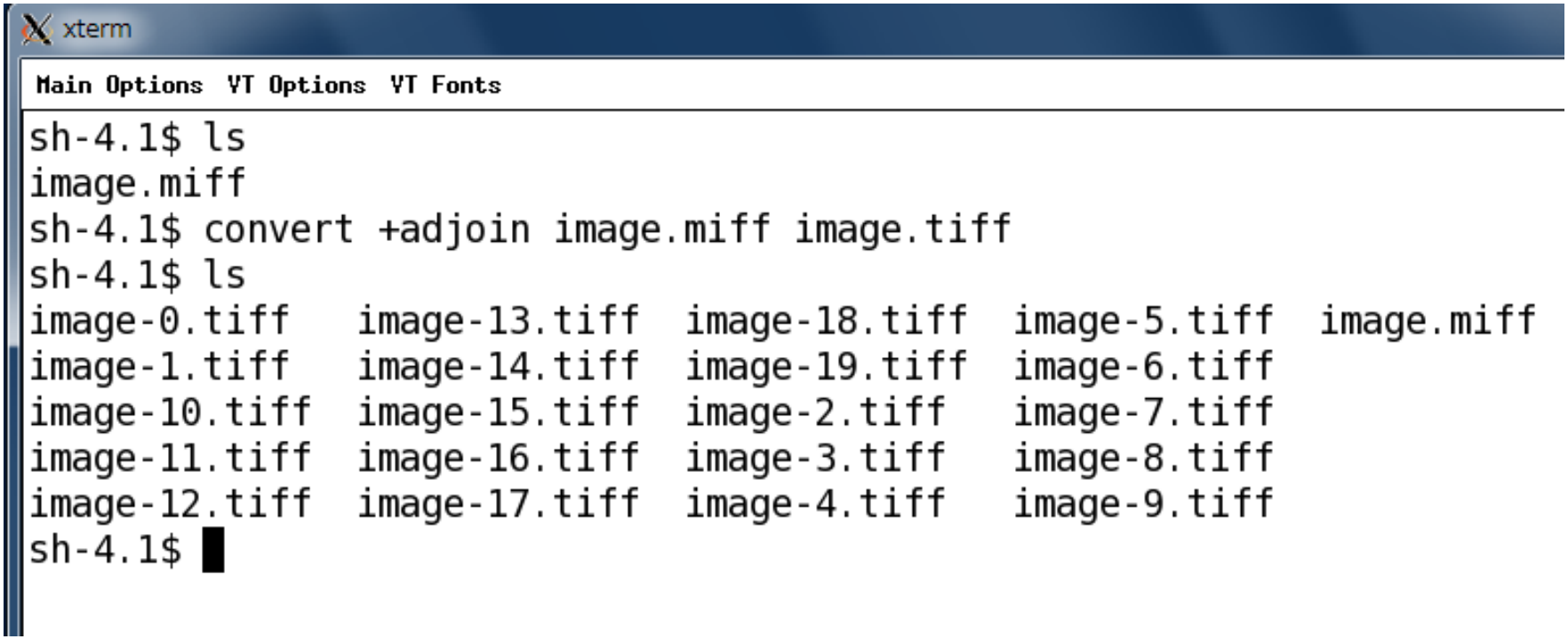
Miff形式のファイルは、Linuxのconvertコマンドで、連番画像に分解すことができる。

convert␣+adjoin␣image.miff␣image.tiff⏎

-> image-0.tiff, image-1.tiff …. が生成される。

Windows用のImageMagick(フリー)を
インストールすれば、Windowsでもこのコマンドが使えるようになる。

Appendix B-11



convertコマンドはCygwinで使用可能