レーザー溶接プログラム

USER'S MANUAL

平成２３年２月

Advanced Algorithm and Systems

目次

1. チュートリアル

1.1 チュートリアル１

最初に基本的な計算手法になれるためにいくつかの簡単な計算をしてみましょう。work ディレクトリの下の tutorial ディレクトリに入ります。ここで ls とすると、

$ ls

boundary.txt　geometry.txt　material.txt　pole_up_dim_1.6mm.txt
chair_dim.txt　input.txt　pole_down_dim.txt　reference.txt
cubic_dim.txt　input_org.txt　pole_down_dim_1.6mm.txt
edge_up_dim_1.6mm.txt　laser_schedule.txt　laser_profile.txt
pole_up_dim.txt

などと表示されます。

このうち計算に直接必要なファイルは

input.txt
reference.txt
boundary.txt
material.txt
geometry.txt
laser_schedule.txt
laser_profile.txt

の７つですが、最後の３つ

geometry.txt
laser_schedule.txt
laser_profile.txt

は、input.txt の設定によっては、必要ありません。その他に

chair_dim.txt
cubic_dim.txt
pole_up_dim_1.6mm.txt
pole_down_dim_1.6mm.txt
edge_up_dim_1.6mm.txt
pole_up_dim.txt
pole_down_dim.txt

等のファイルがありますが、これらは geometry.txt の入力例として用意したファイルで、適宜、名前を geometry.txt に変更して、用います。

現在のディレクトリの状況を確認したところで、実際に計算を実行してみましょう。環境により異なりますが、例えば以下のようにう実行します。

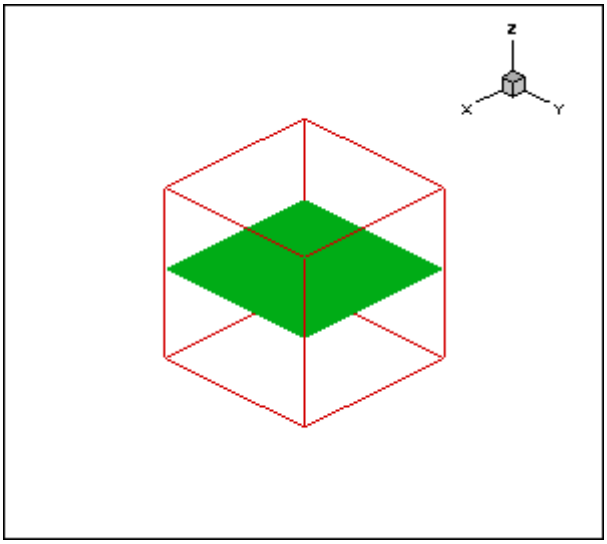
$ ../../Release/welding.exe

```
 num_itr        time(sec)      num_fs          div_v              div_v/div_pv
dp_L2/dp_B2                  v_max             dt_L2                 dt_Diff
avr_tmp              v/v0
     10          0.00001           0      0.00000E+00  NaN              NaN
0.00000E+00          0.11521E+00          0.11265E+00          0.14999E+04
0.10000E+01
     20          0.00002           0      0.00000E+00  NaN              NaN
0.00000E+00          0.10501E+00          0.10242E+00          0.14998E+04
0.10000E+01
     30          0.00003           0      0.00000E+00  NaN              NaN
0.00000E+00          0.97885E-01          0.95302E-01          0.14997E+04
0.10000E+01
     40          0.00004           0      0.00000E+00  NaN              NaN
0.00000E+00          0.92691E-01          0.90147E-01          0.14995E+04
0.10000E+01
     50          0.00005           0      0.00000E+00  NaN              NaN
0.00000E+00          0.88737E-01          0.86244E-01          0.14994E+04
0.10000E+01
     60          0.00006           0      0.00000E+00  NaN              NaN
0.00000E+00          0.85615E-01          0.83185E-01          0.14993E+04
0.10000E+01
     70          0.00007           0      0.00000E+00  NaN              NaN
0.00000E+00          0.83078E-01          0.80721E-01          0.14992E+04
0.10000E+01
     80          0.00008           0      0.00000E+00  NaN              NaN
0.00000E+00          0.80957E-01          0.78671E-01          0.14991E+04
0.10000E+01
     90          0.00009           0      0.00000E+00  NaN              NaN
0.00000E+00          0.79140E-01          0.76937E-01          0.14990E+04
0.10000E+01
    100          0.00010           0      0.00000E+00  NaN              NaN
0.00000E+00          0.77570E-01          0.75438E-01          0.14989E+04
0.10000E+01
```

などと表示されて計算が終了します。ディスプレイに表示される内容については後ほど述べます。まず現在のディレクトリの内容を確認してみましょう。

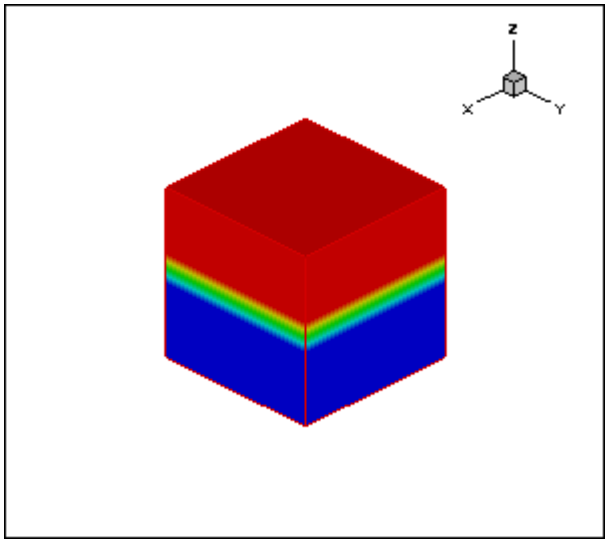
```
$ ls
```

SDF000100.data　　sdf000100.bin
Temperature000100.data　　mass_fraction000100.bin
sdf_prms000100.bin　　pressure000100.bin
temperature000100.bin　　velocity000100.bin
restart000100.txt
上記のファイルが出力されていることがわかります。このうち
SDF000100.data
Temperature000100.data
は表示用の Tecplot 出力ファイルです。それ以外の
sdf000100.bin
mass_fraction000100.bin
sdf_prms000100.bin
pressure000100.bin
temperature000100.bin
velocity000100.bin
restart000100.txt
は計算を改めて再スタートさせたいときに必要になるファイルです。上記.data ファイルがなくてもこれら.bin ファイルと restart*****.txt があれば計算は再開できますし、出力用のファイルも再現できますので、多くの場合これらだけ保存しておけばいいです。（今回、モニタがちゃんと決まっていなかったので、仕方なくこのようにしました。本来であれば、モニタに合わせて圧縮したファイルをこのような目的に使用すべきですので、また次の機会にということで。）
SDF000100.data を表示させてみますと、以下のようになります。

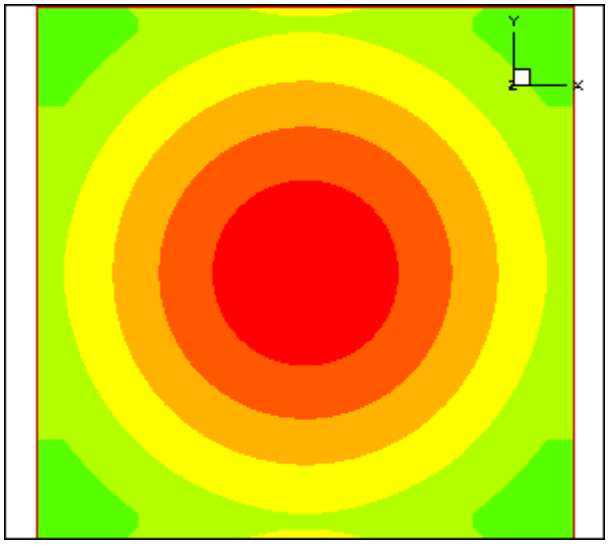


これは signed distance 関数の 0 等値面を出力したものですが、これが気相と固液相の境界

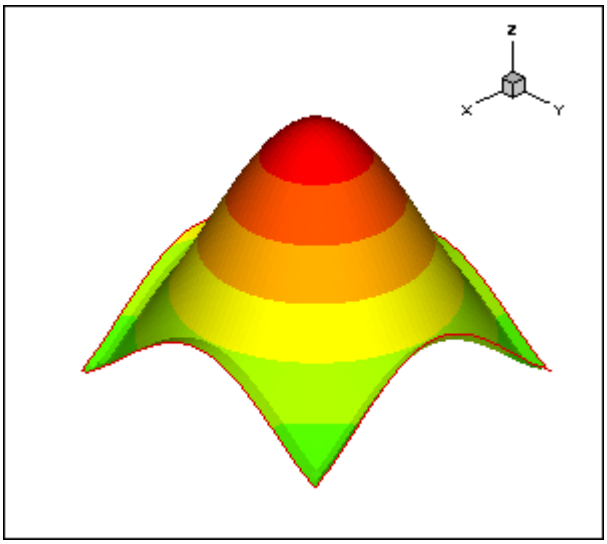
を表します。また、以下に signed distance 関数の正の部分を赤く、負の部分を青く色分けして表示します。



正の赤い部分が、気相をあらわします。また負の青い部分が、固液相を表します。初期条件におけるこれらの気相、固液相界面の設定は簡単なものは、input.txt で設定できます。複雑なものは、geometry.txt に所定のフォーマットに従って設定していただけます。
固液相の内部でどのように溶液が固体部分から溶けて出てくるか、あるいは溶液部分がどのように固体に凝固していくか等については他のファイル、LiquidMassFraction、あるいは SolidMassFraction をモニタするとわかります。これらはフラグひとつで出力できます。
つぎに温度を見てみましょう。



これは気相と固相の界面の温度分図です、 z 軸の上方から見た図です温度を縦軸にとって温度プロファイルを見た図が以下です。



z 軸上方より下方に向けて 1kW の Gaussian beam を照射した結果としてセンタの温度が上昇している様子がわかります。Laser beam は各種の設定ができ、また時間とともに設定を変化させたり、制御することができます。それらは、laser_schedule.txt、laser_profile.txt 等を用いて行います。今行ったの計算では最高温度が 1567.87K ですのでまだ Solidus temperature に達していません。ですからまだ溶融は起きていません。次のチュートリアルでは、今回の計算結果を利用して再スタートし、液相が生じる様子を確認してみましょう。

1.2 チュートリアル２

チュートリアル１で使用したディレクトリ（work/tutorial）に入ります。必要があれば、コピーを別のディレクトリに保存しておくと、何か手違いが起きたときにすぐに復旧できます。

前回計算は、100 回の iteration まで終了しています。100 回目の計算から再出発するためには、チュートリアル１で説明した７つのファイル以外に、以下のファイルが必要です。

sdf000100.bin
mass_fraction000100.bin
sdf_prms000100.bin
pressure000100.bin
temperature000100.bin
velocity000100.bin
restart000100.txt

これらがあることを確認したら、まず restart000100.txt を restart.txt に copy します。例えば以下のようにします。

$ cp　restart000100.txt　restart.txt

次に、input.txt を編集します。

start_condition = initial

と記されている部分を以下のように書き換えてください。

start_condition = restart

次に

number_of_iteration = 100

と記されている部分を以下のように書き換えてください。

number_of_iteration = 5000

計算回数は 5000 回にまで拡張されます。

次に

interval_write = 100

と記されている部分を以下のように書き換えてください。

interval_write = 1000

1000 回ごとに計算結果を出力します。

次に

write_liquid_mass_fraction = .false.

と記されている部分を以下のように書き換えてください。

write_liquid_mass_fraction = .true.

LiquidMassFraction*****.data が出力されます。

次に

include_surface_tension = .false.

と記されている部分を以下のように書き換えてください。

include_surface_tension = .true.

表面張力の効果を計算に組み込みます。以上の変更が終了したら、計算を実行します。例えば以下のようにして計算を実行します。

$ ../../Release/welding.exe

num_itr	time(sec)	num_fs	div_v	div_v/div_pv
110	0.00011	0	0.00000E+00 NaN	NaN
120	0.00012	0	0.00000E+00 NaN	NaN
130	0.00013	0	0.00000E+00 NaN	NaN
140	0.00014	0	0.00000E+00 NaN	NaN
150	0.00015	0	0.00000E+00 NaN	NaN
160	0.00016	0	0.00000E+00 NaN	NaN
170	0.00017	0	0.00000E+00 NaN	NaN
180	0.00018	0	0.00000E+00 NaN	NaN

190	0.00019	0	0.00000E+00 NaN	NaN
200	0.00020	0	0.00000E+00 NaN	NaN
210	0.00021	0	0.00000E+00 NaN	NaN
220	0.00022	0	0.00000E+00 NaN	NaN

・・・・・・

計算の初期においては速度の計算を行っていないため、NaN などが出力されています。これは溶解が始まる前から速度場も含めて計算すると肝心な溶融状態の計算に入るまでの、固体の温度を溶解温度まで持っていくまでに、膨大な計算時間を要してしまうことから、気相が固相の温度上昇に影響する等の事実はひとまず無視して、温度場の計算だけを、溶解が始まるまでは、行うというやり方をデフォルトとして採用していることによります。他の方法として、より近似の強い解析解を初期値として与えてやるという方法もあるでしょう。

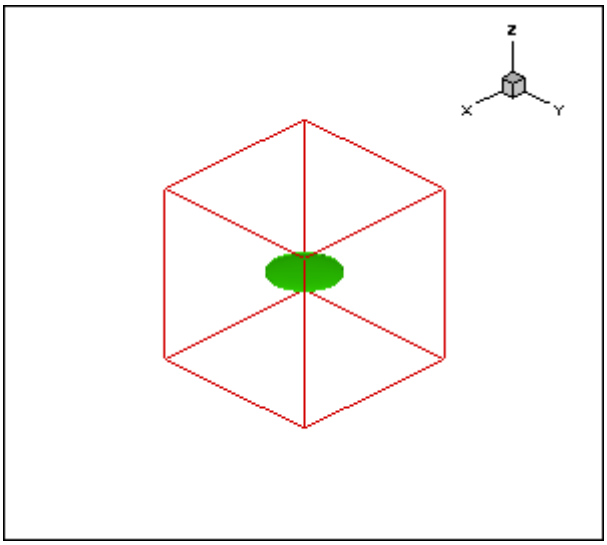
450	0.00045	0	0.00000E+00 NaN	NaN
460	0.00046	0	0.00000E+00 NaN	NaN
470	0.00047	0	0.00000E+00 NaN	NaN
480	0.00048	0	0.00000E+00 NaN	NaN
490	0.00049	0	0.00000E+00 NaN	NaN
500	0.00050	0	0.00000E+00 NaN	NaN
510	0.00051	0	0.00000E+00 NaN	NaN
520	0.00052	0	0.00000E+00 NaN	NaN
530	0.00053	0	0.00000E+00 NaN	NaN
540	0.00054	0	0.00000E+00 NaN	NaN
550	0.00055	0	0.50799E-03	0.48499E+00
560	0.00056	0	0.84761E-03	0.31388E+00
570	0.00057	0	0.11050E-02	0.21969E+00
580	0.00058	0	0.17205E-02	0.19828E+00
590	0.00059	0	0.25822E-02	0.18494E+00
600	0.00060	0	0.38619E-02	0.17664E+00
610	0.00061	0	0.56849E-02	0.17361E+00
620	0.00062	0	0.79804E-02	0.16819E+00
630	0.00063	0	0.10624E-01	0.16099E+00
640	0.00064	0	0.13885E-01	0.15426E+00
650	0.00065	0	0.17837E-01	0.14781E+00
660	0.00066	0	0.30004E-01	0.18856E+00
670	0.00067	0	0.40543E-01	0.19298E+00

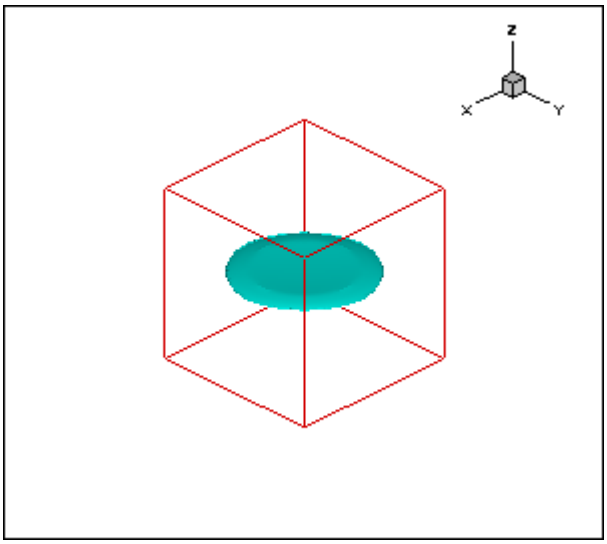
以上を見ると、iteration=540 から iteration=550 の間で、速度場計算が ON していることがわかります。この間に固相の最高温度が、eutectic temperature の 1766.0K あるいは solidus temperature に達したことがわかります。ここで fluid solver の収束判定について 2，3 考えてみましょう。まず、ディスプレイに表示している、div_v は発散 $div\bar{u}$ の絶対値の空間における最大値を無次元量で表現したものです。多くの場合、気相の速度が著しく変化するような領域で生じ、時々刻々生じる場所は変化します。今計算で採用している criterion はかなり甘いものですが、その感覚を得るために、次元化してみましょう。発散は時間の逆数の次元を持っていて、本計算における代表時間は 10.0sec ですので、今回の発散の criterion は

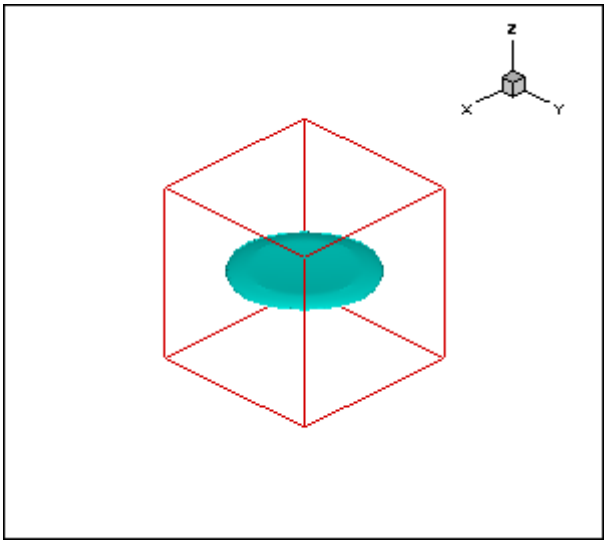
$10^{-2}\,s^{-1}$

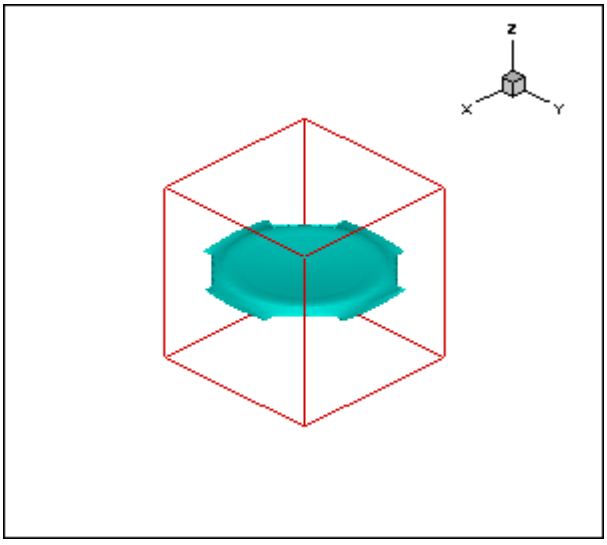
となります。これは、もしこの値を発散が同じ空間のある地点において 1sec の間維持したら、その空間セル内の 1 %の mass が損なわれるか、あるいは付け加わってしまうよ、という類の量です。多くの実用計算でははるかに強い拘束を課すのですが、気相と固液相の計算ではあまり実用的ではありません。いくつかのいいわけをするならば、つねに固液相の発散は、気相のそれよりひとけたから、ふたけた値が小さいこと、空間の大部分では、もっと小さい発散量を維持していること、発散の criterion を著しく上げてもそれほど計算結果に差が出ないことが多いなどがあげられますが、より良い値が得られることが好ましいことに相違ありません。また今回は reinitialization においてはもちろんのこと（実はナイーヴに計算した場合には、reinitialization で上述の何百倍ものマスが失われます。）level set の発展方程式においてもマスを完全に保存する拘束を課すことができるようになっています。正当性については、計算結果を待つしかないように思います。

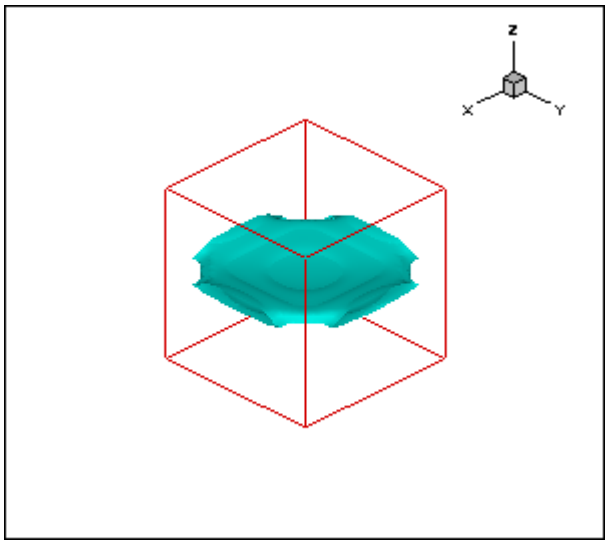
では、計算結果を見ていきましょう。




Z
X
Y


Z
X
Y


Z
X
Y



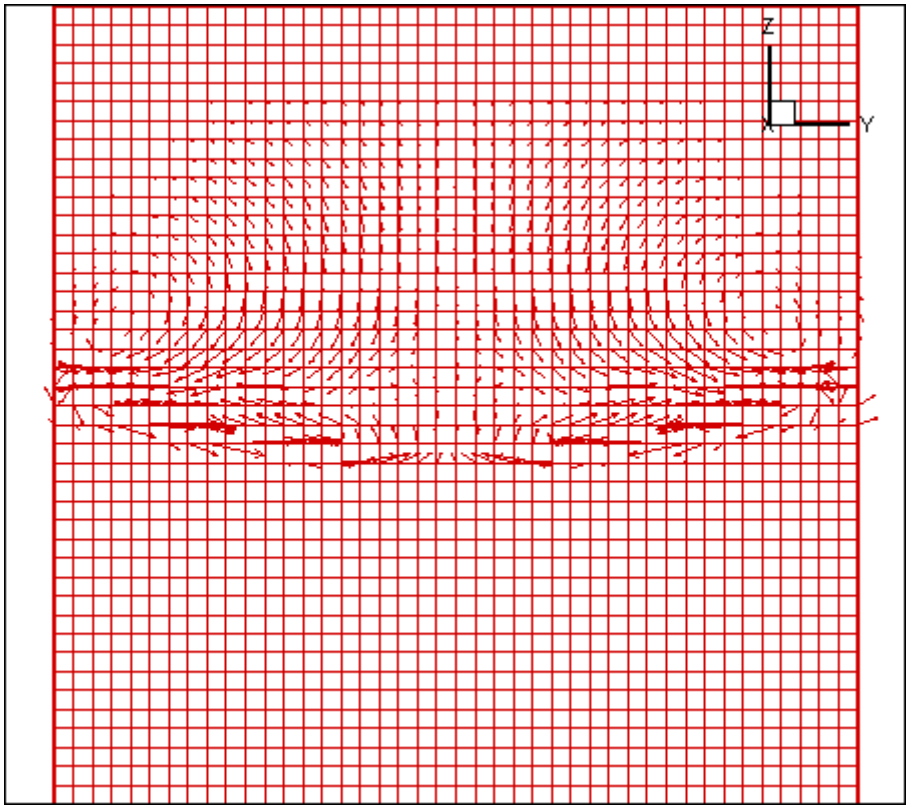
以上は time=0.001sec から time=0.005sec の間の liquid mass fraction の 0.25 等値面です。温度が最も早く上昇する、中心部で溶解が始まり、徐々に回りに広がっていきます。
次に速度場を確認してみましょう。先ほど input.txt を編集した際に速度場を出力するフラグを立てなかったので、速度場は出力されていないはずです。最初から計算しなおす必要はありません。まず、restart.txt を以下のように変更します。

$ cp　restart005000.txt　　restart.txt

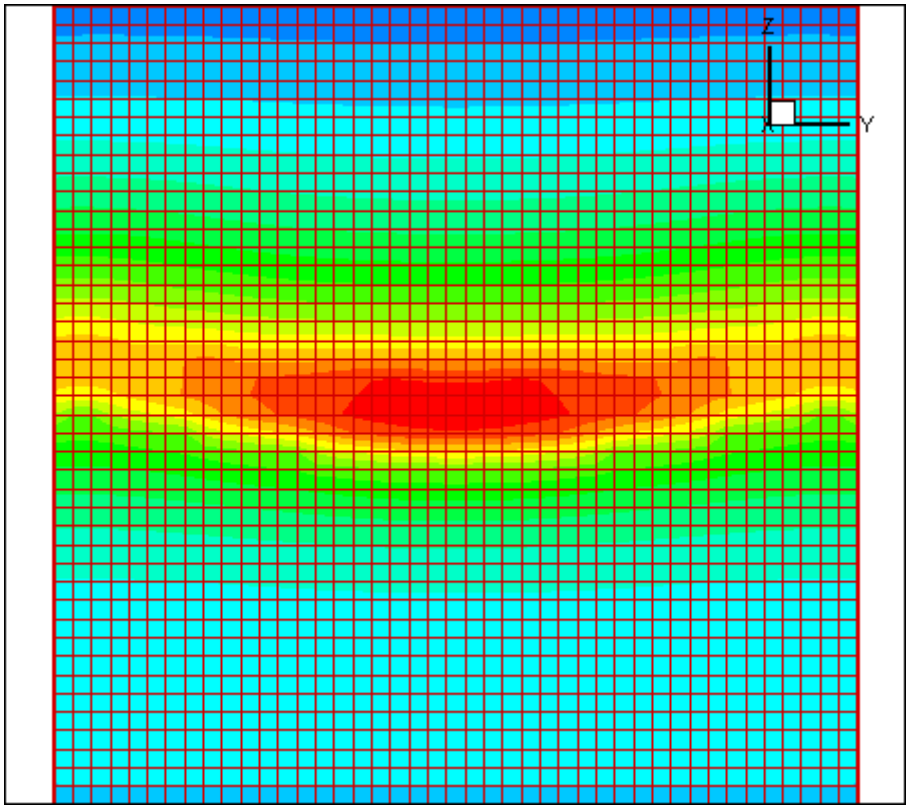
次に input.txt を以下のように修正します。
write_velocity　　= .false.
を
write_velocity　　= .true.
と修正する。
これで、例えば以下のように計算を実行すると、
$ ../../Release/welding.exe
 num_itr　　time(sec)　　num_fs　　div_v　　div_v/div_pv

と、計算がすぐに終了します。VelocityS005001.data を表示すると、以下のようになります。表面張力が温度の勾配によりドライヴされている様を観察できます。今回の物質は温度勾配係数が負ですので高温領域から、低音領域へ溶融物が流されています。つまり、表面においては、中心部から周辺部へ溶融物は流れ、内部領域では逆向きに、周辺部から中心部へ向けて流れが生じています。

速度場



温度場



1.3　チュートリアル３

本チュートリアルでは、geometry.txt を簡単に編集して、少し複雑な形状を laser beam で加熱溶解してみましょう。まず work/tutorial3 に移動しましょう。本題に入る前に、計算領域について説明します。計算領域は、実際に以下の物理場や補助関数が計算される、空間領域で、直方体をしています。

計算領域で定義される物理量

Velocity
Pressure
Temperature
Liquid mass fraction
Solid mass fraction
Signed distance function

計算領域は、input.txt の

nx	=	40	
ny	=	40	
nz	=	40	
dx	=	0.00004	# [m]
dy	=	0.00004	# [m]
dz	=	0.00004	# [m]

（現在のところ dx=dy=dz しかサポートしていません。）

の６つの entry から以下の８点を定義して、それから張られる直方体として定義できます。

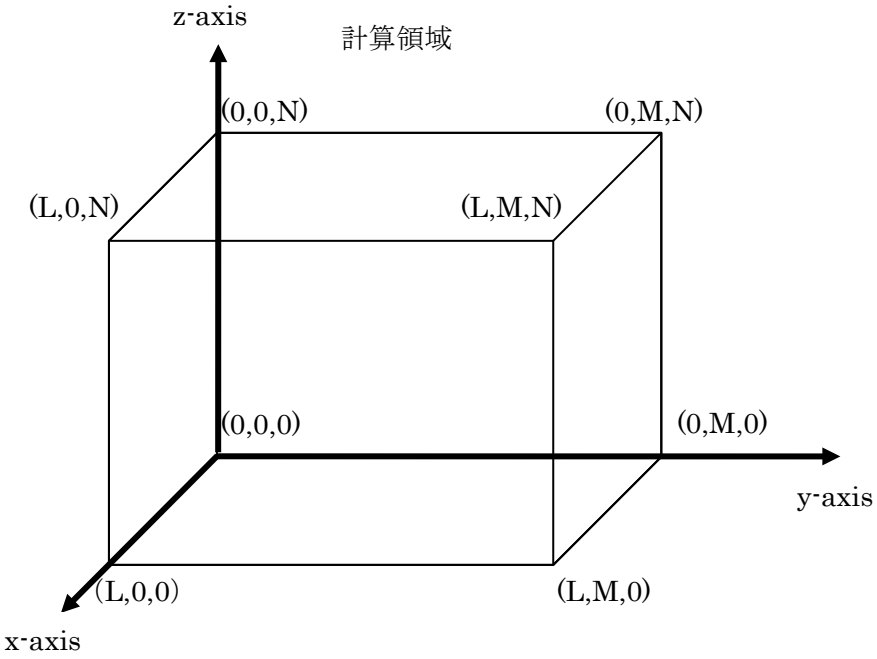
$$(0,0,0),(L,0,0),(0,M,0),(L,M,0),(0,0,N),(L,0,N),(0,M,N),(L,M,N)$$

ここで

$L = nx \times dx \quad (m)$

$M = ny \times dy \quad (m)$

$N = nz \times dz \quad (m)$



今回行う計算では、

$L = 40 \times 0.00004 = 0.0016$

$M = 40 \times 0.00004 = 0.0016$

$N = 40 \times 0.00004 = 0.0016$

ですので、一辺が 1.6mm の立方体が計算領域になっています。

この計算領域に、様々な形の物質を配置するのが geometry.txt です。geometry.txt のフォーマットは以下のようになります。

num_polygon

x_1 y_1 z_1

x_2 y_2 z_2

x_3 y_3 z_3

x_4 y_4 z_4

……

……

$x_{num\_polygon\times 3}$ $y_{num\_polygon\times 3}$ $z_{num\_polygon\times 3}$

細かい三角形をいくつか配置して形状を表現します。

x_1 y_1 z_1

x_2 y_2 z_2

x_3　　y_3　　z_3

が最初の三角形の３つの空間座標を表します。以下

x_4　　y_4　　z_4

x_5　　y_5　　z_5

x_6　　y_6　　z_6

が２番目の三角形の３つの空間座標を、

x_7　　y_7　　z_7

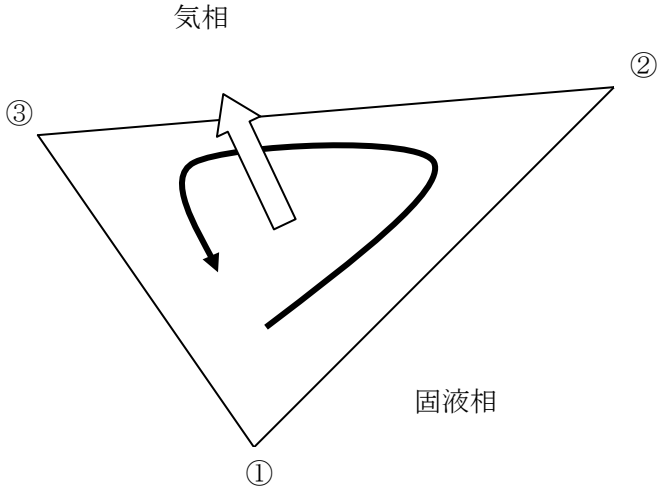
x_8　　y_8　　z_8

x_9　　y_9　　z_9

が３番目の三角形の３つの空間座標を表します。

三角形は好きなだけ配置することができます。

三角形には向きがあって、右ねじの向きに（座標が記述された順番に従って右ねじを回したときに右ねじが進む方向）signed distance 関数は正の値をとります。そしてそれは気相が配置される側になります。



例１

$z = 0.8mm$ を通って、ｚ軸に垂直な平面で気相－物質界面をつくり、$z > 0.8mm$ を気相で満たし、$z < 0.8mm$ を物質で満たしたいときは、

2

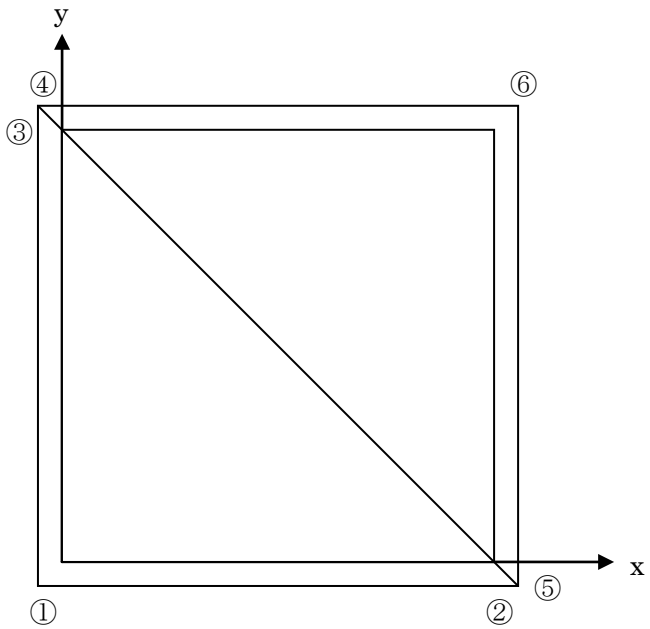
-0.0001　　-0.0001　　0.0008

 0.0017　　-0.0001　　0.0008

-0.0001　　 0.0017　　0.0008

-0.0001 0.0017 0.0008
0.0017 -0.0001 0.0008
0.0017 0.0017 0.0008

などとします。やり方は一通りではありません。



それでは実際にためしてみましょう。
上の例に従って geometry.txt を編集してください。あるいは、example1.txt に同じことがすでに記されていますので、それを geometry.txt にコピーしてください。
次に input.txt の関係箇所を書き換えます。

sdf_initial_mode = plane # plane, sphere, file_input

となっている部分を以下のように書き換えます。

sdf_initial_mode = file_input # plane, sphere, file_input

#以降はコメント文なのでコンピュータには読み込まれません。これで、signed distance 関数が、geometry.txt から読み込まれて初期化されます。signed distance 関数が出力されるよう、

write_signed_distance = .true.

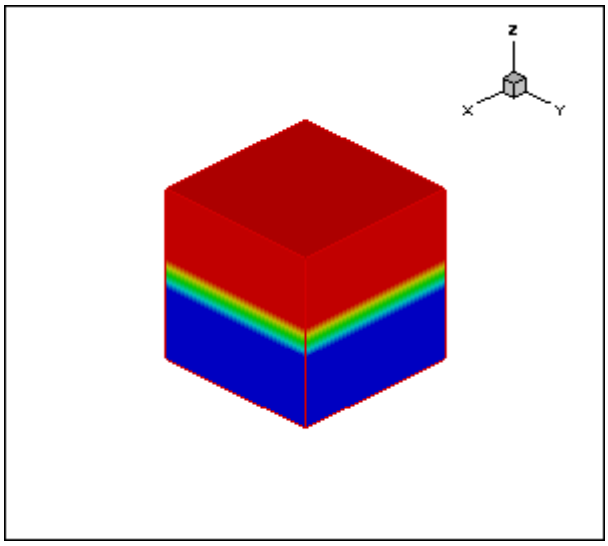
となっていることを確認したらソルバーを実行します。

$ ../../Release/welding.exe

num_itr	time(sec)	num_fs	div_v	div_v/div_pv	
10	0.00001	0	0.00000E+00	NaN	NaN
20	0.00002	0	0.00000E+00	NaN	NaN

30	0.00003	0	0.00000E+00 NaN	NaN
40	0.00004	0	0.00000E+00 NaN	NaN
50	0.00005	0	0.00000E+00 NaN	NaN
60	0.00006	0	0.00000E+00 NaN	NaN
70	0.00007	0	0.00000E+00 NaN	NaN
80	0.00008	0	0.00000E+00 NaN	NaN
90	0.00009	0	0.00000E+00 NaN	NaN
100	0.00010	0	0.00000E+00 NaN	NaN

計算が終了したら、正しく物質領域が定義されているか、SDF000100.data をチェックしてみましょう。



図では赤い部分が正の部分、青い部分が負の部分になっていますので、正しく領域が配置されたことがわかります。

〔追記〕この例にあるぐらいの物質形状だと、わざわざ geometry.txt を編集しなくても input.txt を以下のように編集しても同じことができます。これらに関する詳細な説明は、入力ファイルの説明を行う章で改めて、行います。

```
sdf_initial_mode        =       plane                   # plane, sphere, file_input
sdf_initial_radius      =       0.010                   # [m]
sdf_initial_center      =       0.0008  0.0008  0.0008  # [m]
sdf_initial_direction   =       0.0     0.0     1.0
```

以下の例を実際にやってみる場合には、

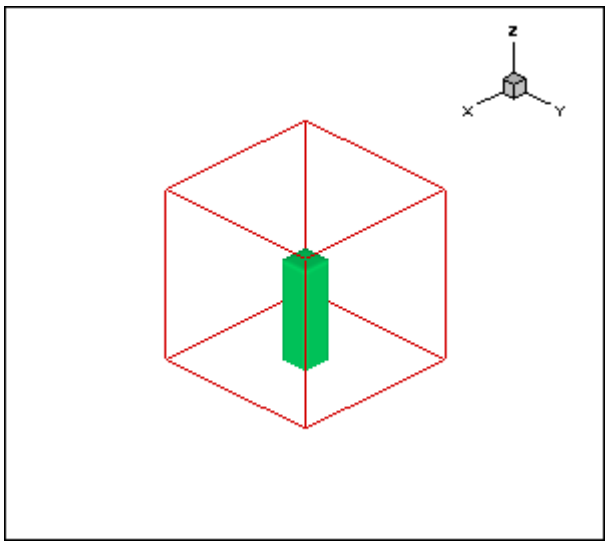
```
sdf_initial_mode        =       file_input
```

になっていることを、しっかり確認してください

例 2　　pole_up_dim_1.6mm.txt

pole_up_dim_1.6mm.txt を geometry.txt に copy してソルバーを実行します。

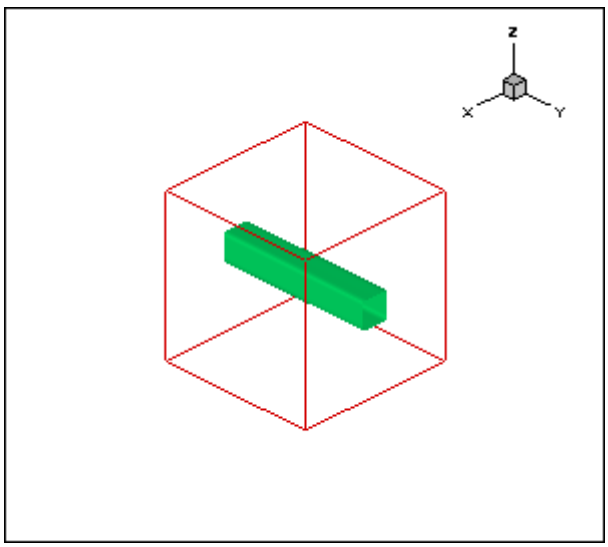
以下の物質形状が得られます。



例 3　　pole_lay_dim_1.6mm.txt

pole_lay_dim_1.6mm.txt を geometry.txt に copy してソルバーを実行します。

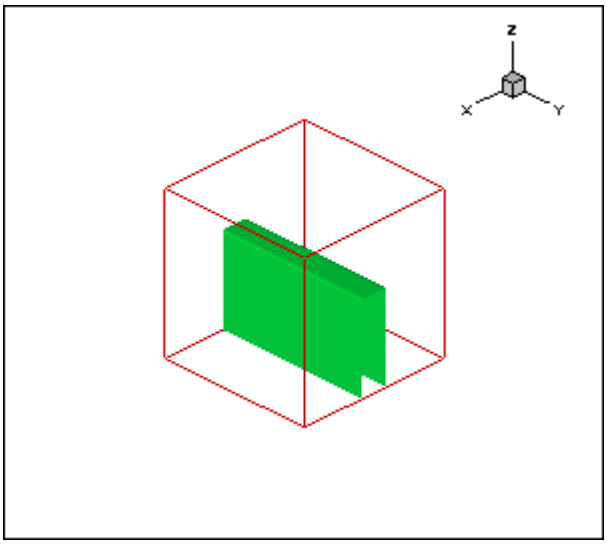
以下の物質形状が得られます。



例 4　　edge_up_dim_1.6mm.txt

edge_up_dim_1.6mm.txt を geometry.txt に copy してソルバーを実行します。

以下の物質形状が得られます。



1.4　チュートリアル 4

本チュートリアルでは、レーザービームの制御の仕方を学習します。今回、実装したのは、

Gaussian beam

$$\tilde{u}(x,y,z)=\left(\frac{2}{\pi}\right)^{\frac{1}{2}}\frac{\exp[-ikz+i\psi(z)]}{w(z)}\exp\left[-\frac{x^2+y^2}{w^2(z)}-ik\frac{x^2+y^2}{2R(z)}\right]$$

Hermite-Gaussian Mode

$$\tilde{u}_{nm}(x,y,z)=\left(\frac{2}{\pi}\right)^{\frac{1}{2}}\frac{\exp[i(m+n+1)\psi(z)]}{\sqrt{2^m m!\,w(z)}\sqrt{2^n n!\,w(z)}}H_m\left(\frac{\sqrt{2}x}{w(z)}\right)H_n\left(\frac{\sqrt{2}y}{w(z)}\right)\exp\left[-ikz-ik\frac{x^2+y^2}{2R(z)}\right]$$

$$\exp\left[-\frac{x^2+y^2}{w^2(z)}\right]$$

Laguerre-Gaussian Mode

$$\tilde{u}_{pm}(r,\theta,z)=\sqrt{\frac{2p!}{\pi(m+p)!}}\frac{\exp[i(2p+m+1)(\psi(z)-\psi_0)]}{w(z)}\left(\frac{\sqrt{2}r}{w(z)}\right)^m L_p^m\left(\frac{2r^2}{w^2(z)}\right)$$

$$\exp\left[-ikz-ik\frac{r^2}{2R(z)}-im\theta-\frac{r^2}{w^2(z)}\right]$$

ここで

$$w(z) = w_0\sqrt{1+\left(\frac{z}{z_R}\right)^2}$$

$$R(z) = z + \frac{{z_R}^2}{z}$$

$$\psi(z) = \tan^{-1}\left(\frac{z}{z_R}\right)$$

$$z_R = \frac{1}{2}k{w_0}^2 = \frac{\pi}{\lambda}{w_0}^2$$

です。

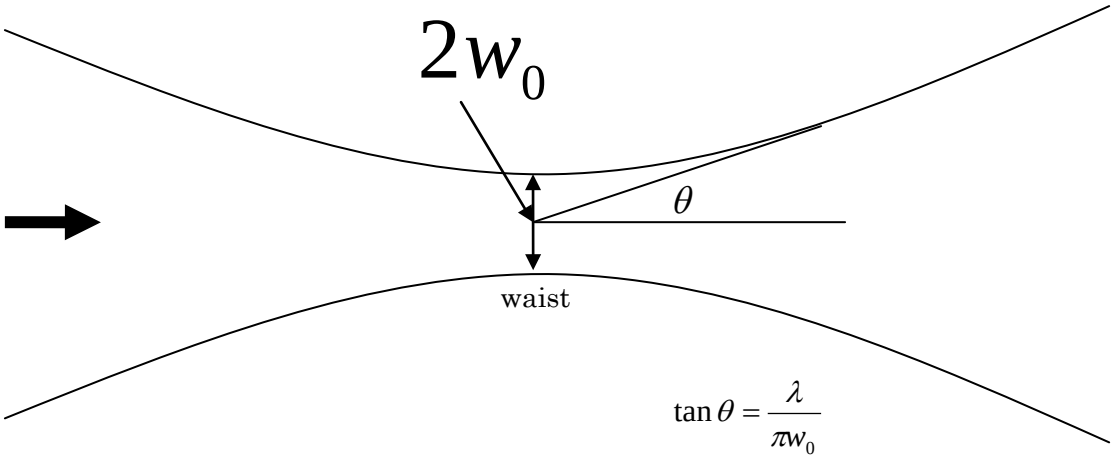
Gaussian beam の profile は waist size w_0 と、beam の波長 λ だけで決まることがわかります。以上の 3 つは wave equation の paraxial approximation から導出されますが、以下のファイルから読み込む任意形状用の laser beam はそれよりもいい加減なもので、wave equation の解にはなっていませんが、便宜上の理由で実装しました。

ファイルからの任意形状

$$\tilde{u}(x,y,z) = \Pr(x,y)\frac{\exp[-ikz + i\psi(z)]}{w(z)}\exp\left[-ik\frac{x^2+y^2}{2R(z)}\right]$$

ここで $\Pr(x,y)$ は適当に規格化された形状関数です。

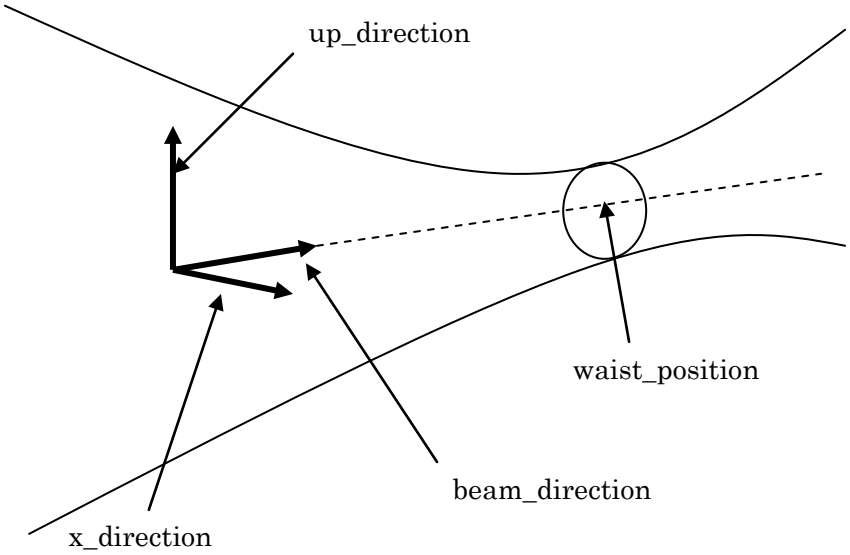
以下に Gaussian beam の伝播の様子を示します。



beam が最も絞れているところを waist(ウェスト)と呼びます。waist における beam 幅が大体 $2w_0$ です。w_0 が小さければ小さいほど、また beam の波長 λ が、大きければ大きいほど遠方で beam は大きく広がります。input.txt を見てみると、beam を特徴づける entry として以下のものがあります。

```
# Laser beam parameters
include_laser_beam                =       .true.
use_laser_schedule                =       .false.
laser_wave_length                 =       1.064e-6                    # [m]
power_transmitted_by_the_beam     =       200                         # [W]
waist_size                        =       1.2e-3                      # [m]
waist_position                    =       0.0008    0.0008    0.0008  # [m]
beam_direction                    =       0.0   0.0   -1.0             # [non dim]
up_direction                      =       0.0   1.0   0.0       # [non dim]
laser_mode                        =       Gaussian          # Gaussian
                                                              # Hermite_Gaussian
                                                              # Laguerre_Gaussian
                                                              # file_input
laser_order_m                     =       3
laser_order_n                     =       3
times_diffraction_limited         =       5
quality_factor                    =       1.2
```

ここではまず、waist_position、beam_direction、up_direction の説明をします。
waist_position は beam を最も絞った点の中心の座標を表します。beam_direction は laser_beam のエネルギーの流れる方向を表す、方向ベクトルです。Hermite-Gaussian Mode や任意形状プロファイルでは、beam_direction に垂直な面内における座標を特定する必要が生じますが、up_direction は beam に垂直な面内において y 軸方向を定義するのに用いられます。この y 軸は、前のチュートリアルで定義した、空間領域を定義するために用いた座標とは異なります。



それでは実際に計算してこれらのパラメータの意味を確認してみましょう。まず、計算領域は、前チュートリアルの例１と同じものを選びます。つまり、1.6mm 四方の立方体の空間領域に、0.8mm までの高さの物質を満たします。そのために、input.txt の関連箇所が以下のようになっていることを確認しましょう。

```
sdf_initial_mode        =    plane                  # plane, sphere, file_input
sdf_initial_center      =    0.0008  0.0008  0.0008   # [m]
sdf_initial_direction   =    0.0     0.0     1.0
```

次に、この物質に対して鉛直上方から（z 軸の正の側から）z 軸の負の方向へ向けて、$w_0 = 0.2mm$ の Gaussian beam を照射しましょう。また up_direction は実際の y 軸と一致するように選びます。input.txt の関連箇所が以下のようになっていることを確認しましょう。

include_laser_beam = .true.

レーザービームを計算に加えることを指定します。

laser_wave_length = 1.064e-6 # [m]

レーザーの波長を指定します。

power_transmitted_by_the_beam = 200 # [W]

レーザーが総量として単位時間当たりのエネルギー量を指定します。

waist_size = 0.2e-3 # [m]

上で与えた式における、w_0 の値を指定します。

waist_position = 0.0008 0.0008 0.0008 # [m]

ビームのウェスト位置を計算領域の中央に指定します。

beam_direction = 0.0 0.0 -1.0 # [non dim]

ビームの照射方向を z 軸の負の方向に決めます。

up_direction = 0.0 1.0 0.0 # [non dim]

ビームの照射方向に垂直な面内における y 方向を計算領域の y 方向と一致させます。

laser_mode = Gaussian

Gaussian beam を照射します。

以下のように温度の計算結果が出力されるよう設定されていることを確認しましょう。

write_temperature = .true.

それでは計算を実行しましょう。

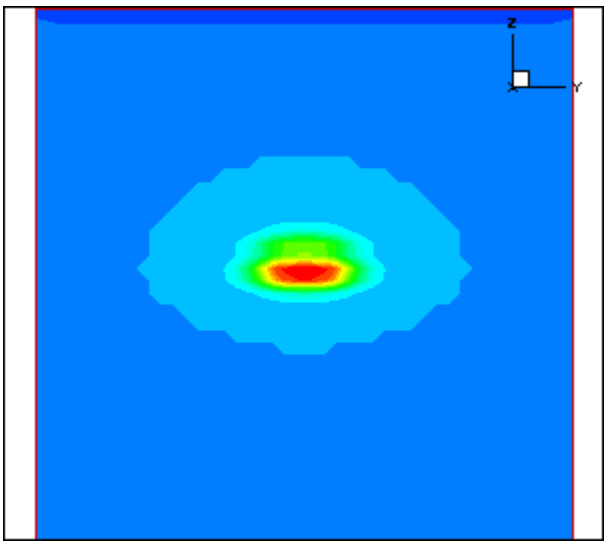
$ ../../Release/welding.exe

num_itr	time(sec)	num_fs	div_v	div_v/div_pv	
10	0.00001	0	0.00000E+00	NaN	NaN
20	0.00002	0	0.00000E+00	NaN	NaN
30	0.00003	0	0.00000E+00	NaN	NaN
40	0.00004	0	0.00000E+00	NaN	NaN
50	0.00005	0	0.00000E+00	NaN	NaN
60	0.00006	0	0.00000E+00	NaN	NaN
70	0.00007	0	0.00000E+00	NaN	NaN
80	0.00008	0	0.00000E+00	NaN	NaN
90	0.00009	0	0.00000E+00	NaN	NaN
100	0.00010	0	0.00000E+00	NaN	NaN

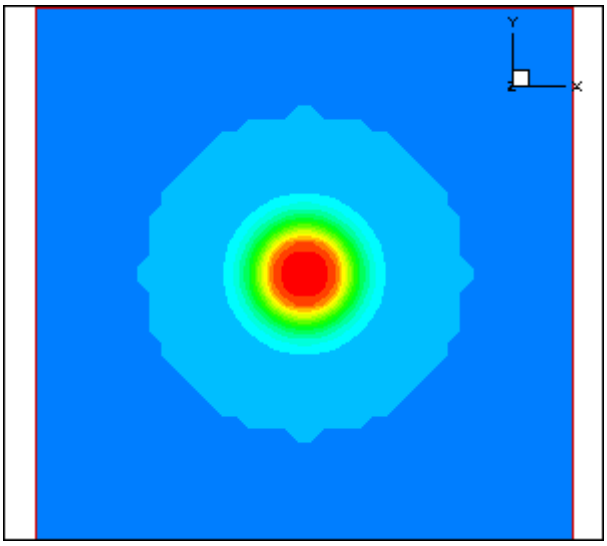
計算が終了したら、温度分布を Temperature000100.data などで確認してみましょう。

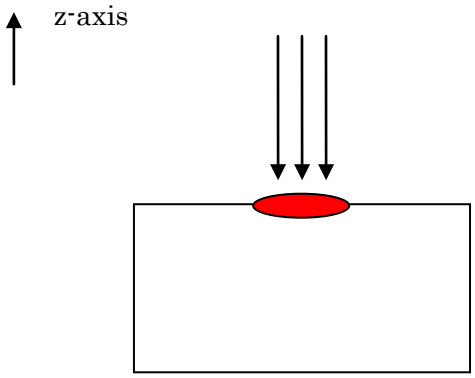
x 軸の正の方向から見た温度分布では、物質の界面がレーザーにより照射されて温度が上昇している様子を確認できます。また、レーザーの照射方向である、z 軸の正の方向から見た温度分布では、レーザーが界面に垂直に照射し Gauss 分布らしく完全な円対称が確認できます。

x 軸の正の方向から見た温度分布



z 軸の正の方向から見た温度分布





次に up_direction の意味がわかりやすい例を考えて見ましょう。そのために、laser_mode として Hermite_Gaussian を選択します。以下のように、input.txt を編集してください。

```
laser_mode              =       Hermite_Gaussian
laser_order_m           =       3
laser_order_n           =       2
```

これで beam に local な座標系で x 軸方向のオーダが 3、y 軸方向のオーダが 2 の Hermite-Gaussian beam が先ほどと同じ方向へ照射されることになります。それでは、実行してみましょう。

```
$ ../../Release/welding.exe
 num_itr        time(sec)       num_fs          div_v                   div_v/div_pv
      10          0.00001            0          0.00000E+00 NaN                    NaN
      20          0.00002            0          0.00000E+00 NaN                    NaN
      30          0.00003            0          0.00000E+00 NaN                    NaN
      40          0.00004            0          0.00000E+00 NaN                    NaN
      50          0.00005            0          0.00000E+00 NaN                    NaN
      60          0.00006            0          0.00000E+00 NaN                    NaN
      70          0.00007            0          0.00000E+00 NaN                    NaN
      80          0.00008            0          0.00000E+00 NaN                    NaN
      90          0.00009            0          0.00000E+00 NaN                    NaN
     100          0.00010            0          0.00000E+00 NaN                    NaN
```

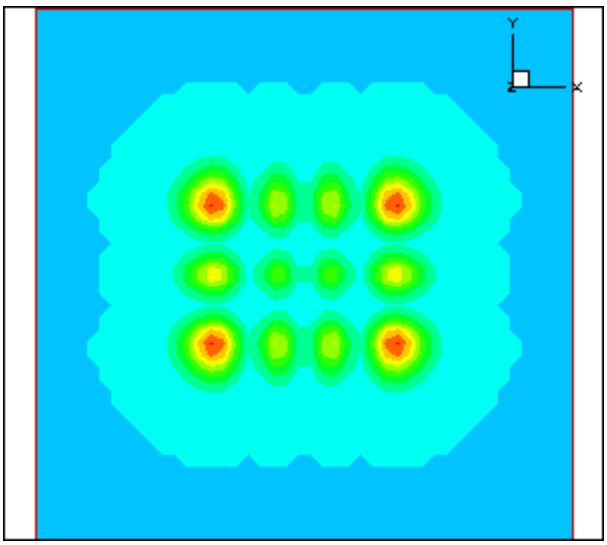
計算が終了したら、温度分布を Temperature000100.data などで確認してみましょう。
z 軸の正の方向から物質界面の温度分布を観察してみると、x 軸方向に 3 つの零点が、y 軸方向に 2 つの零点が確認できると思います。それでは次に、input.txt の up_direction を以下のように書き直してもう一度計算を実行してみましょう。

```
up_direction                            =       1.0     0.0     0.0
```

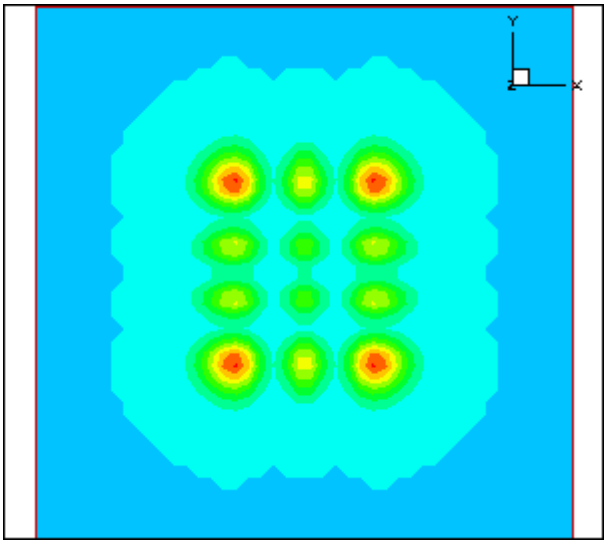
Hermite-Gaussian Mode の実際のビームの幅はおおよそ以下のように広がります。

$$\approx 2\sqrt{n}w(z)$$

up_direction=(0.0,　1.0,　0.0)の時の表面の温度分布



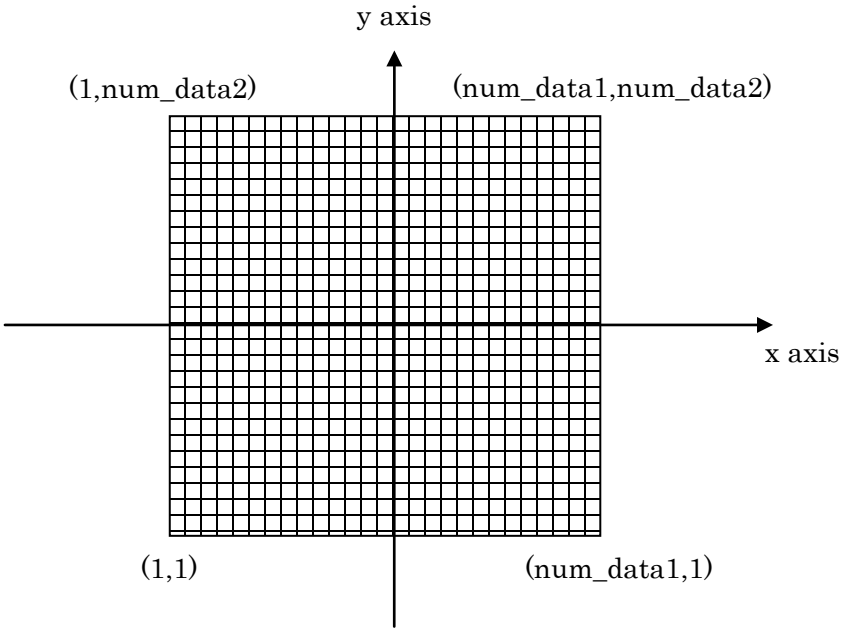
up_direction=(１.0,　0.0,　0.0)の時の表面の温度分布



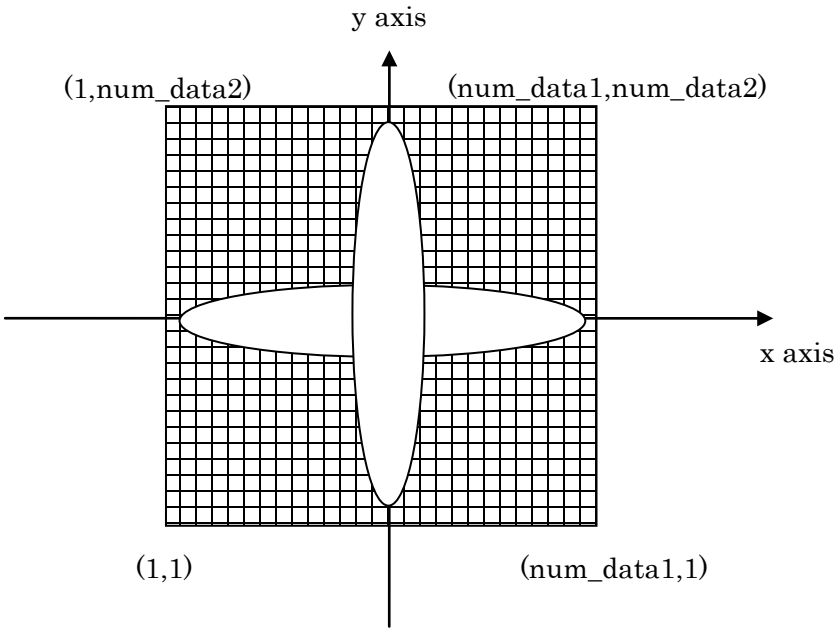
本チュートリアルの最後に、ファイルからビームのプロファイルを読み込む形の、レーザービーム照射について学習しましょう。この mode はあまり正確ではありませんが、いろいろなビーム形状をとにかく試してみたいというときに便利かもしれません。
ビームのプロファイルを設定するには、laser_profile.txt を編集します。laser_profile.txt のフォーマットは以下のようになります。

num_data1　　num_data2
data(1,1)
data(2,1)
data(3,1)

data(num_data1,num_data2)



ここで num_data1 と num_data2 は必ずしも等しくとる必要はありませんが、どちらも奇数を指定してください。data(i,j)には図に示した、対応する点のレーザ強度を記述してください。強度は、相対的なものでいいです。内部で適切に規格化を行います。では、例として以下の強度分布を持つレーザビームの、laser_profile.txt を作成してみましょう。



まず、以下のフォートランプログラムを作成します。

```
program main
    implicit none
    integer:: i,j,n
    real:: x,y,d,int

    n=21
    d=2.0/(n-1)
    write(6,'(2i5)'),n,n
    do j=1,n
        y=-1.0+d*real(j-1)
        do i=1,n
            x=-1.0+d*real(i-1)
            int=exp(-x**2-(y*10.0)**2)+exp(-(x*10.0)**2-y**2)
            write(6,'(e19.6)') int
        end do
    end do

end program main
```

上述のプログラムを例えば、beam_profile.exe という名前でコンパイルしてから、以下の

ように実行します。

$ ../Debug/laser_profile.exe　>　laser_profile.txt

すると laser_profile.txt が作成されますが、こうして作られた、laser_profile.txt が tutorial4 の中にすでに入っています。これだけではまだソルバーは laser_profile.txt を読みにいってくれません。input.txt に以下の修正を加えます。

laser_mode　=　Hermite_Gaussian

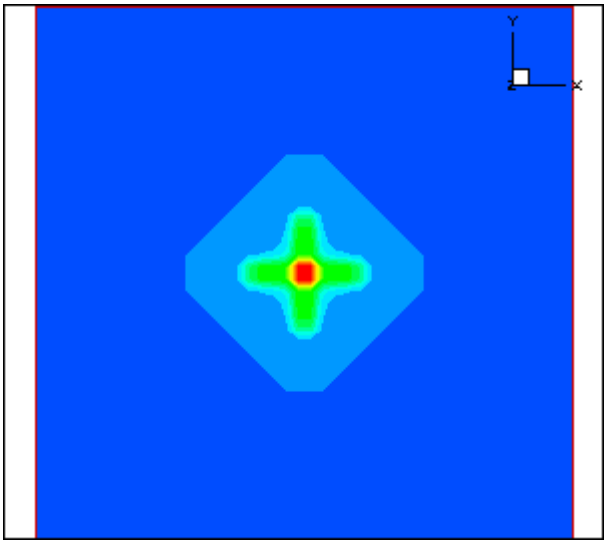
とあるのを以下のように修正してください。

laser_mode　=　file_input

これで、ソルバーを実行できます。

$ ../../Release/welding.exe

num_itr	time(sec)	num_fs	div_v	div_v/div_pv	
10	0.00001	0	0.00000E+00	NaN	NaN
20	0.00002	0	0.00000E+00	NaN	NaN
30	0.00003	0	0.00000E+00	NaN	NaN
40	0.00004	0	0.00000E+00	NaN	NaN
50	0.00005	0	0.00000E+00	NaN	NaN
60	0.00006	0	0.00000E+00	NaN	NaN
70	0.00007	0	0.00000E+00	NaN	NaN
80	0.00008	0	0.00000E+00	NaN	NaN
90	0.00009	0	0.00000E+00	NaN	NaN
100	0.00010	0	0.00000E+00	NaN	NaN



z 軸の正の方向からみた温度分布

1.5　チュートリアル５

今回はレーザービームを走査させるなど、時間とともにレーザーのパラメータを変化させる方法を学びます。そのために使うのが、laser_schedule.txt です。フォーマットは以下の通りです。

number_of_data

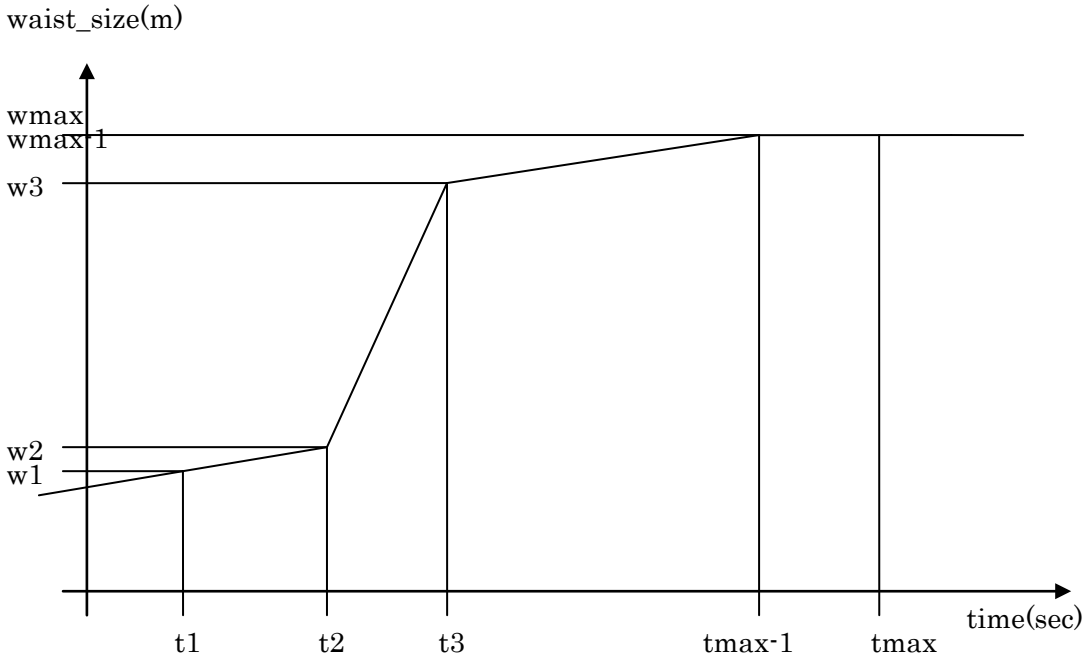
time1(sec)　power1(Wat)　waist_size1(m)　waist_position1(m)(x,y,z)…….

time2(sec)　power2(Wat)　waist_size2(m)　waist_position2(m)(x,y,z) …….

time3(sec)　power3(Wat)　waist_size3(m)　waist_position3(m)(x,y,z) …….

.....

waist_size を例にとると



時刻 t1 以前の値は、時刻 t1 の値 w1 と時刻 t2 の値 w2 を線形に外挿して決めます。

時刻 t　$t_i \le t < t_j$　の値は時刻 ti の値 wi と時刻 tj の値 wj を線形内挿して決めます。

時刻 $t_{\max}$ 以降の値は時刻 $t_{\max-1}$ の値 $w_{\max-1}$ と時刻 $t_{\max}$ の値 $w_{\max}$ を線形に外挿して決めます。

それでは実際にレーザービームを走査した例を計算してみましょう。まず、計算領域は、前チュートリアルと同じものを選びます。つまり、1.6mm 四方の立方体の空間領域に、0.8mm までの高さの物質を満たします。そのために、input.txt の関連箇所が以下のようになっていることを確認しましょう。

```
sdf_initial_mode          =      plane                  # plane, sphere, file_input
sdf_initial_center        =     0.0008   0.0008   0.0008     # [m]
sdf_initial_direction     =       0.0      0.0      1.0
```

次に laser_schedule.txt を以下のように編集します。

```
5
0.0000    200     0.2e-3     0.0002 0.0002 0.0008     0.0 0.0 -1.0     0.0 1.0 0.0
0.0001    200     0.2e-3     0.0008 0.0014 0.0008     0.0 0.0 -1.0     0.0 1.0 0.0
0.0002    200     0.2e-3     0.0014 0.0002 0.0008     0.0 0.0 -1.0     0.0 1.0 0.0
0.00021   200     0.2e-3     0.0005 0.0008 0.0008     0.0 0.0 -1.0     0.0 1.0 0.0
0.0003    200     0.2e-3     0.0011 0.0008 0.0008     0.0 0.0 -1.0     0.0 1.0 0.0
```

これだけでは、ソルバーは laser_schedule.txt を読みにいかないので、input.txt の以下の部分を修正します。

```
use_laser_schedule                =      .false.
```

を

```
use_laser_schedule                =      . true.
```

さらに、以下の部分を修正してください。

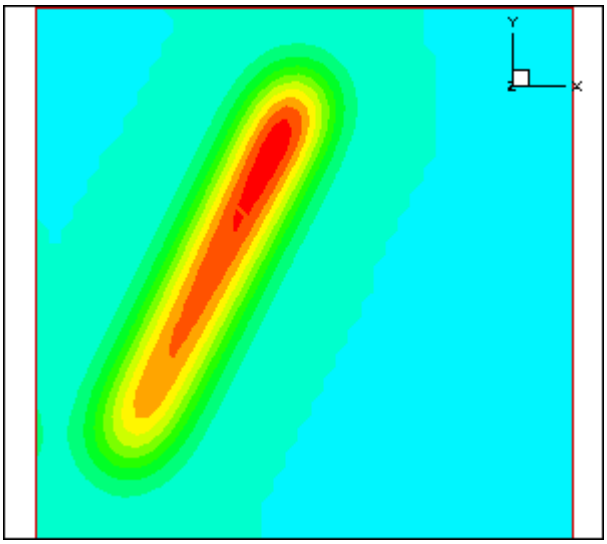
```
number_of_iteration       =       100
```

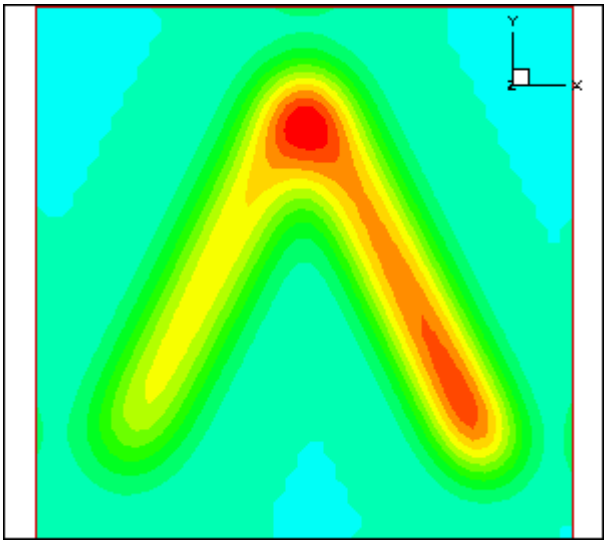
を

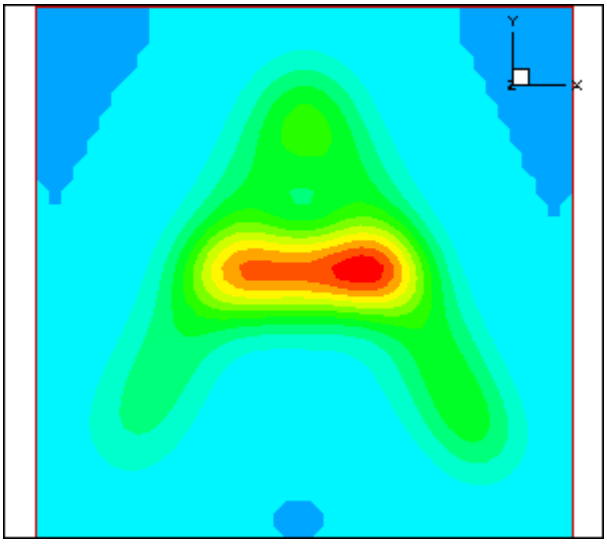
```
number_of_iteration       =       300
```

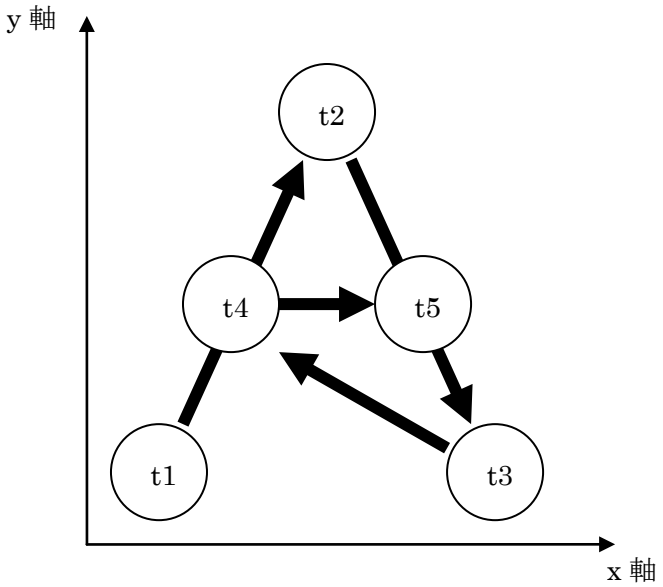
それでは準備ができたので、計算してみましょう。

```
$ ../../Release/welding.exe
 num_itr        time(sec)     num_fs          div_v              div_v/div_pv
    10           0.00001          0       0.00000E+00 NaN                  NaN
    20           0.00002          0       0.00000E+00 NaN                  NaN
    30           0.00003          0       0.00000E+00 NaN                  NaN
    40           0.00004          0       0.00000E+00 NaN                  NaN
    50           0.00005          0       0.00000E+00 NaN                  NaN
    60           0.00006          0       0.00000E+00 NaN                  NaN
. . . . . .
   260           0.00026          0       0.00000E+00 NaN                  NaN
   270           0.00027          0       0.00000E+00 NaN                  NaN
   280           0.00028          0       0.00000E+00 NaN                  NaN
   290           0.00029          0       0.00000E+00 NaN                  NaN
   300           0.00030          0       0.00000E+00 NaN                  NaN
```


Y
Z
X


Y
Z
X


Y
Z
X



現在の実装では、ray tracing 等を完全に実施しているわけではないので、実行上、不自然な結果が出る可能性が、状況によっては起こりえますが、今後の課題にしておきたいと思います。

2　入力ファイルの説明

2.1　input.txt

nx　：　計算領域の x 方向の分割数。
ny　：　計算領域の y 方向の分割数。
nz　：　計算領域の z 方向の分割数。

dx　：　x 方向のグリッド幅。単位はメートル。
dy　：　y 方向のグリッド幅。単位はメートル。
dz　：　z 方向のグリッド幅。単位はメートル。

現在の実装では、dx=dy=dz を仮定していて、それ以外は指定できません。
計算領域は、以下の８点により張られる直方体として定義できます。

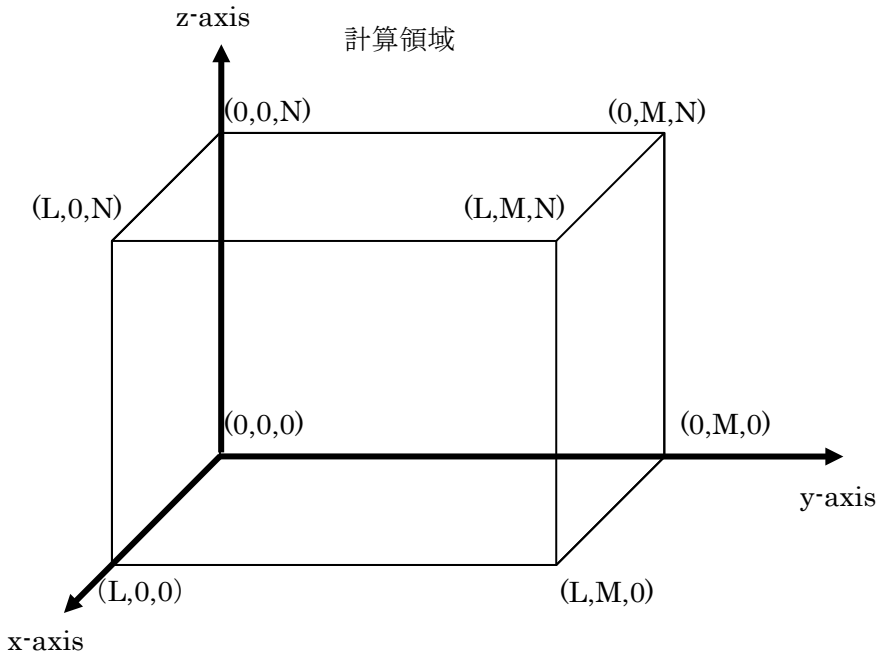
$(0,0,0),(L,0,0),(0,M,0),(L,M,0),(0,0,N),(L,0,N),(0,M,N),(L,M,N)$

ここで

$L = nx \times dx$ (m)

$M = ny \times dy$ (m)

$N = nz \times dz$ (m)



dt　：計算時間ステップ幅。単位は sec、秒。

interval_of_velocity_calculation　：　速度の計算は、dt×interval_of_velocity_calculation ごとに、タイムステップ dt×interval_of_velocity_calculation で行われます。温度の計算が律速になってタイムステップが取れない場合に、実行速度を速めるために使用します。十分にタイムステップが取れる場合には、1 を指定すると最も精度が保障されます。

start_condition　：　initial を指定すると指定した初期条件にしたがって、時刻 0.0 から計算が開始されます。restart を指定すると、restart.txt の内容に従って、計算が再実行されます。

initial_temperature　：　計算開始時の温度をケルヴィン単位で指定します。

initial_velocity　：　計算開始時の速度を MKS 単位で指定します。

number_of_iteration　：　計算するステップの総数を指定します。

end_time　：　計算する総時間を秒単位で指定します。

number_of_iteration あるいは end_time のどちらか一方の条件が満たされたときに計算は終了します。

interval_display　：　ディスプレイに表示される計算状況の報告を何 iteration 毎に行うかを指定します。

interval_write　：　モニタ用のデータおよび、再計算用のデータの出力を、何 iteration 毎に行うかを指定します。

omega_temperature　：　流体ソルバーを SCGZ にした場合のオーバーリラクゼーション係数だが、現在のところ使用していません。

include_gravity　：　.true.を指定すると、計算に重力を加えます。.false.を指定すると重力は計算しません。

gx　：　重力の方向ベクトルの x 成分。単位ベクトルで指定します。
gy　：　重力の方向ベクトルの y 成分。単位ベクトルで指定します。
gz　：　重力の方向ベクトルの z 成分。単位ベクトルで指定します。

calc_Buoyancy　：　.true.を指定すると、温度変化に伴う、密度変化を考慮して、牛力計算を行います。.false.を指定すると、温度変化に伴う、密度変化を考慮しません。

include_surface_tension　：　.true.を指定すると、表面張力を計算に組み込みます。.false.を指定すると、表面張力は計算しません。

surface_tension_shift　：　気相とそれ以外の相の界面は、signed distance 関数の値＝0 で定義しますが、実際の物性は滑らかな Heaviside 関数により定義しています。そのため、界面における諸量をそれに付随した滑らかなδ関数で定義してしまうと、本来、例えば、固液相において吸収されるべきとして定義した、レーザエネルギーが気相と固液相に分散して（文字通りδ関数を定義すると丁度半分ずつに分配されてしまいます。）。同様に表面張力も、固液相をドライヴするものとして定義した量が、気相のドライヴにも使われてしまう状況を観測できます。それに対する、対応策として、完全なものではないですが、δ

関数を、幾分固液相にシフトできるように（マイナスを指定すれば、気相側へもシフトできます。）指定できるようにしました。単位はグリッドの幅（dx=dy=dz）です。0 を指定いただければ、従来の δ 関数で計算することができます。

solid_velocity　：　固相の速度を指定します。0 以外の値を指定するときには、接する外部境界の速度も矛盾しないように設定します。

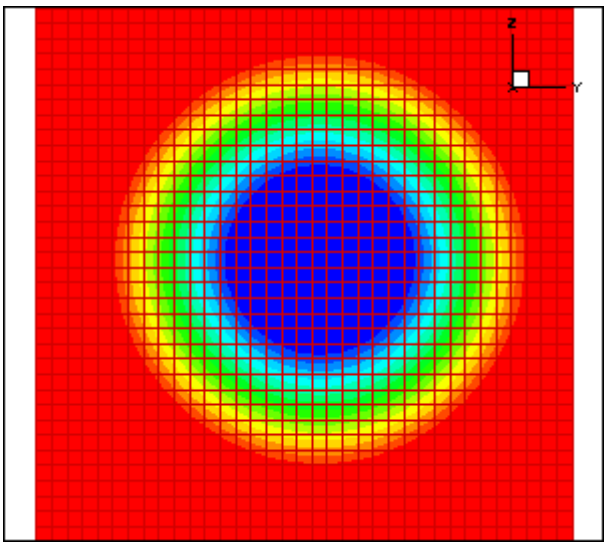
darcy_epsilon　：　mushy zone における Darcy force の特異点を除くための定数です。

$$\frac{1}{g_\ell^{\,3}} \quad -> \quad \frac{1}{g_\ell^{\,3} + \varepsilon_{Darcy}}$$

laser_power　：　現在、未使用
spatial_laser_distribution　：　現在、未使用

Signed distance function parameters
sdf_width　：　$\pm dx \times$(sdf_width)の範囲で signed distance 関数を計算します。それ以外の範囲では、正の場合には、$+dx \times$(sdf_width)の値を、負の場合には、$-dx \times$(sdf_width)の値を sdf に代入します。以下に 3 次元空間内の球の例を示します。

sdf_width=4.0 の場合の球の signed distance 関数の例



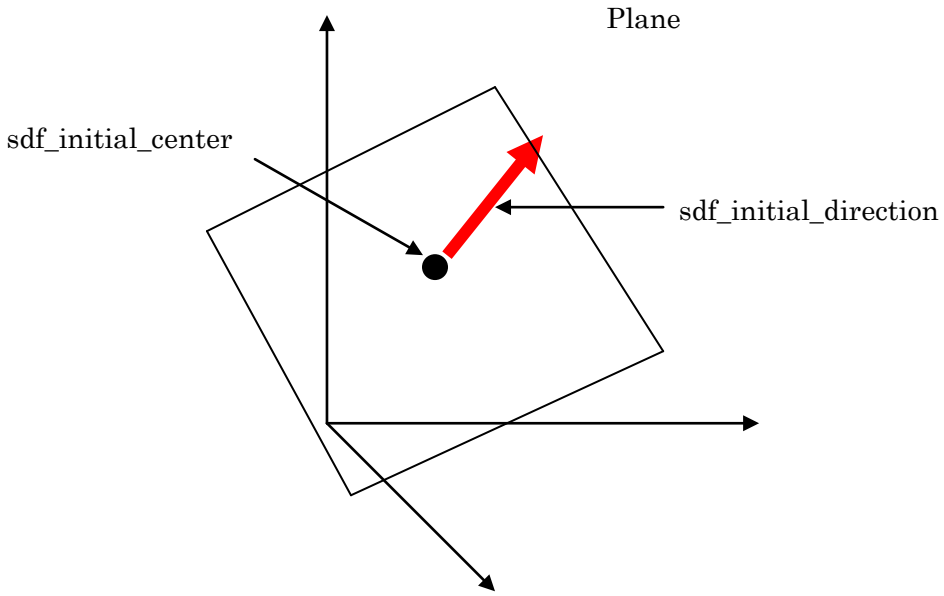
heaviside_epsilon　：　物性を定義するための Heaviside 関数は以下で定義されます。

$$H(\phi)=\begin{cases}0 & \phi<-\varepsilon \\ \frac{1}{2}+\frac{\phi}{2\varepsilon}+\frac{1}{2\pi}\sin\left(\frac{\pi\phi}{\varepsilon}\right) & -\varepsilon\le\phi\le\varepsilon \\ 1 & \varepsilon<\phi\end{cases} \qquad (3)$$

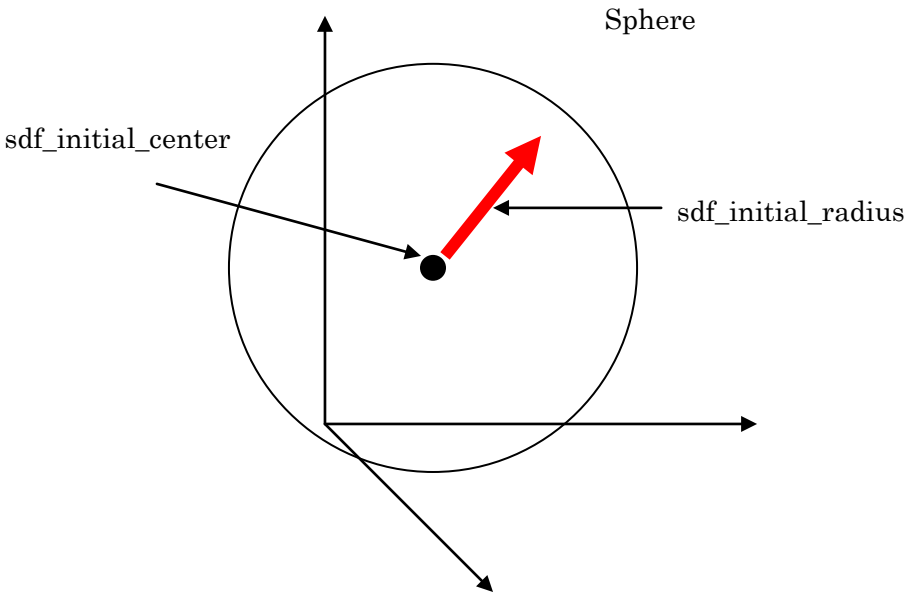
ここでϕは signed distance function です。heaviside_epsilon は上の式におけるεです。

sdf_initial_mode　：　sined distance 関数の初期値を設定します。plane,sphere,file_input の３種類が設定できます。

sdf_initial_mode = plane のときは、初期 sdf として平面が指定されます。
sdf_initial_center　：　平面が通る点の座標を指定します。単位はメートルです。
sdf_initial_direction　：　平面の法線ベクトルを指定します。法線ベクトルは、sdf 関数の負の方向から正の方向へ向かうように指定します。また、これは、固液相から気相へ向かう方向でもあります。



sdf_initial_mode = sphere のときは、初期 sdf として球面が指定されます。
sdf_initial_radius　：　球面の半径を指定します。単位はメートルです。
sdf_initial_center　：　球面の中心の座標を指定します。単位はメートルです。
球面内部が、負の値をとり、球面外部が正の値を取ります。現在のところこれがデフォルトで逆はできません。



sdf_initial_mode = file_input のときは、外部ファイルにより sdf 関数を設定します。外部ファイルとして、geometry.txt を用います。そのフォーマットはチュートリアルに書かれていますが、再度繰り返しますと、

num_polygon

x_1 y_1 z_1

x_2 y_2 z_2

x_3 y_3 z_3

x_4 y_4 z_4

……

……..

$x_{num\_polygon\times3}$ $y_{num\_polygon\times3}$ $z_{num\_polygon\times3}$

細かい三角形をいくつか配置して形状を表現します。

x_1 y_1 z_1

x_2 y_2 z_2

x_3 y_3 z_3

が最初の三角形の 3 つの空間座標を表します。以下

x_4 y_4 z_4

x_5 y_5 z_5

x_6　　y_6　　z_6

が２番目の三角形の３つの空間座標を、

x_7　　y_7　　z_7

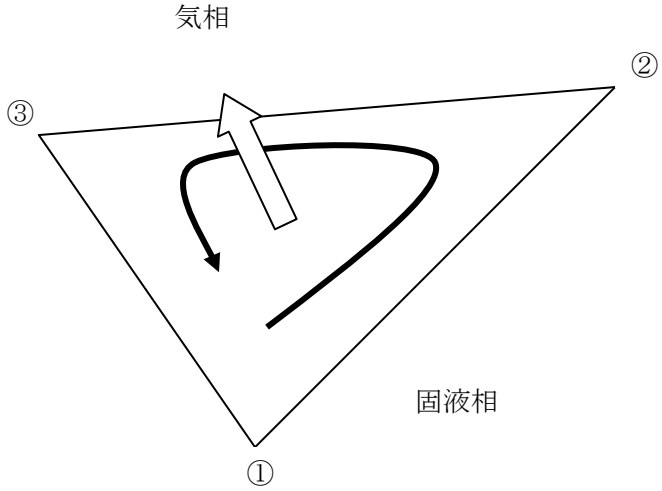
x_8　　y_8　　z_8

x_9　　y_9　　z_9

が３番目の三角形の３つの空間座標を表します。

三角形は好きなだけ配置することができます。

三角形には向きがあって、右ねじの向きに（座標が記述された順番に従って右ねじを回したときに右ねじが進む方向）signed distance 関数は正の値をとります。そしてそれは気相が配置される側になります。



例については、チュートリアルならびに各 tutorial ディレクトリ内の、geometry.txt 等を参照ください。

sdf_itr_max　：　現在、未使用です。SCGZ にて使用します。

sdf_omega　：　現在、未使用です。SCGZ にて使用します。

sdf_epsilon　：　現在、未使用です。SCGZ にて使用します。

level_set_scheme　：　以下の signed distance 関数の時間発展を計算する数値スキームを選択、指定します。

$$\frac{\partial \phi(\vec{r},t)}{\partial t} + \vec{V} \cdot \vec{\nabla} \phi(\vec{r},t) = 0$$

ここで$\phi(\vec{r},t)$は signed distance 関数です。

lss_upwind_1st と lss_upwind_1st_cnstrnt を指定することができます。lss_upwind_1st を指定すると、up-wind の空間一次精度で計算します。lss_upwind_1st_cnstrnt を指定すると、up-wind の空間一次精度に global な mass を保存する拘束条件をつけたものを計算します。ここでいう global な mass とは、

$$\int_{\Omega}[1.0-H(\phi(\vec{r},t))]d\Omega$$

でΩは、計算領域全体を、$H(\phi)$は Heaviside 関数を、$\phi(\vec{r},t)$は signed distance 関数を表します。

redistance_scheme は signed distance 関数、$\phi(\vec{r},t)$を最初期化するための数値スキームを指定します。以下の 6 種類が指定可能です。

rds_upwind_1st

rds_upwind_1st_cnstrnt

rds_fast_marching_1st

rds_eno_3rd

rds_eno_3rd_cnstrnt

rds_weno_5th

rds_upwind_1st を指定すると

$$\frac{\partial\phi(\vec{r},t)}{\partial t}=sign(\phi_0)\left(1.0-\left|\vec{\nabla}\phi(\vec{r},t)\right|\right) \qquad (1)$$

を upwind の空間一次精度で解きます。

rds_upwind_1st_cnstrnt を指定すると、上記、redistancing の方程式（1）を、各グリッドセル内の mass を保存するという拘束条件をつけて解きます。ここでいう、各グリッドセル内の mass とは

$$\int_{\Omega_{ijk}}[1.0-H(\phi(\vec{r},t))]d\Omega_{ijk} \qquad 1\le i\le nx\ 、\ 1\le j\le ny\ 、\ 1\le k\le nz$$

で、Ω_{ijk}は(i,j,k)番目のグリッドセル内の空間領域を、$H(\phi)$は Heaviside 関数を、$\phi(\vec{r},t)$は signed distance 関数を表します。

rds_fast_marching_1st を指定すると、空間一次精度の Fast Marching で redistancing を行います。

rds_eno_3rd を指定すると、eno(essentially non-oscillatory scheme)の空間 3 次精度で上記，redistancing の方程式（1）を解きます。

rds_eno_3rd_cnstrnt を指定すると、上記、redistancing の方程式（1）を、各グリッドセル内の mass を保存するという拘束条件をつけて、eno(essentially non-oscillatory scheme)

の空間 3 次精度で解きます。
rds_weno_5th は現在のところ予約してありますが、未実装です。

Fluid solver parameters
convection_scheme　：　流体ソルバーの convection 項の数値スキームを指定します。以下の 3 つが指定できます。
upwind_c
upwind_nc
eno2nd

upwind_c を指定すると、convection 項に $\vec{\nabla}(\vec{u}u_i)$ の保存型を仮定して、upwind の空間一次で解きます。upwind_nc を指定すると、convection 項を $\vec{u}\cdot\vec{\nabla}u_i$ として、upwind の空間一次で解きます。eno2nd を指定すると、convection 項を $\vec{u}\cdot\vec{\nabla}u_i$ として、eno の空間 2 次で解きます。

max_itr_scgs　：　SCGZ 用に用意した変数ですが、現在のところ未使用です。

max_itr_fs　：　fractional step において pressure を project するときに解く以下の poisson solver の最大繰り返し回数を指定します。

$$\vec{\nabla}\left(\frac{1}{\rho}\vec{\nabla}p\right)=\frac{\vec{\nabla}\cdot\vec{V}^*}{\Delta t} \qquad (2)$$

eps_v　：　式（2）を解く poisson solver の収束条件を下で述べる、fluid_solver_norm を用いて、divergence あるいは、divergence_relative で指定したときの、閾値を与えます。

eps_p　：　式（2）を解く poisson solver の収束条件ををを下で述べる、fluid_solver_norm を用いて、p_res_l2_relative で指定したときの、閾値を与えます。

omega_velocity　：　SCGZ 用に用意した変数ですが、現在のところ未使用です。

omega_pressure　：　式（2）を解く poisson solver の over relaxation を与えます。

fluid_solver_norm　：　式（2）を解く poisson solver の収束条件として何を用いるかを指定します。以下の 3 つを指定できます。

divergence
divergence_relative
p_res_l2_relative
divergence を指定すると、収束条件は以下のようになります。

$$\max_{(i,j,k)}\left|div\vec{u}\right| \quad < \quad \text{eps\_v}$$

divergence_relative を指定すると、収束条件は以下のようになります。

$$\frac{\max_{(i,j,k)}\left|div\vec{u}\right|}{\max_{(i,j,k)}\left|div\vec{u}^{*}\right|} \quad < \quad \text{eps\_v}$$

p_res_l2_relative を指定すると、式（２）を数値的に離散化したものを行列で以下のように表したとき、

$$A\vec{p} = \vec{b}$$

収束条件は以下のようになります。

$$\frac{\left\|\vec{b} - A\vec{p}\right\|^{L2}}{\left\|\vec{b}\right\|^{L2}} \quad < \quad \text{eps\_p}$$

ここで L2-norm $\left\|\bullet\right\|^{L2}$ は以下で定義されるスカラー量です。

$$\left\|\vec{p}\right\|^{L2} = \sqrt{\sum_{(i,j,k)} p_{i,j,k}^{2}}$$

また、行列Ａは対角項が１になるように規格化してとります。

Laser beam parameters

include_laser_beam　：　レーザービームの照射を計算に加えるかどうかを指定するためのフラグです。.true.なら温度の計算にレーザービームの照射は考慮されます。.false.の場合には、レーザービームの照射は考慮されません。

use_laser_schedule　：　レーザービームのパラメータの経時変化を記述するのに、laser_schedule.txt を使用するかどうかを指定するためのフラグです。.true.なら laser_schedule.txt を使用します。.false.なら使用しません。laser_schedule.txt のフォーマットについては、チュートリアル５、もしくは laser_schedule.txt の章を参照してください。

laser_wave_length　：　レーザービームの波長をメートル単位で指定します。

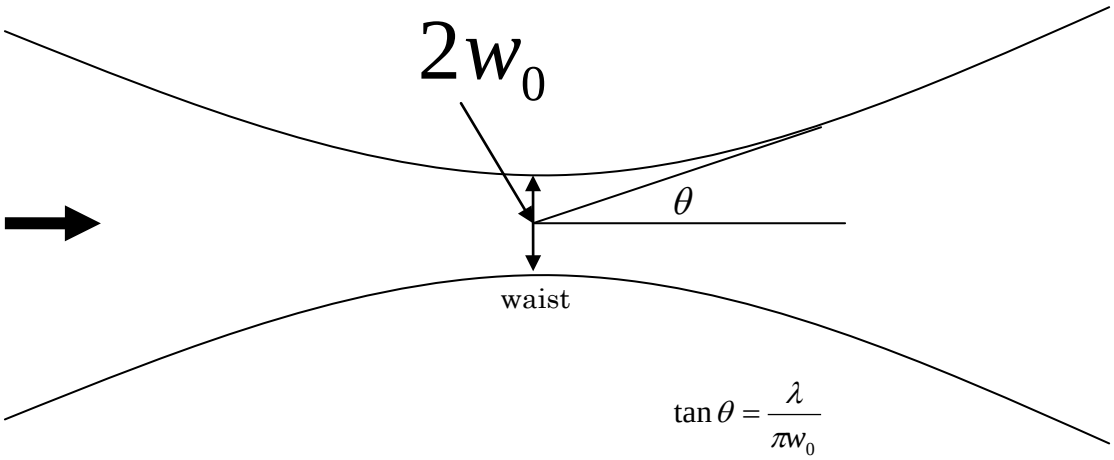
power_transmitted_by_the_beam　：　実際にレーザービームによって単位時間当たり運搬されるエネルギー量を W 単位で指定します。

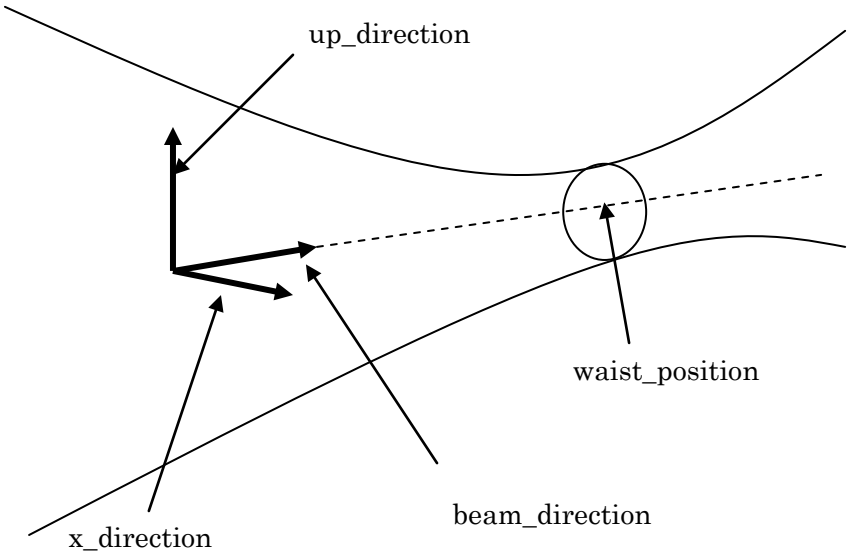
waist_size　：　チュートリアル、あるいは以下で述べる式における、w_0 を表します。Gaussian beam においては最もビームが絞られた点におけるビーム幅がおおよそ $2w_0$ になります。より高次の Herimite-Gaussian beam においては最もビームが絞られた点におけるビーム幅がおおよそ x 方向で、$2\sqrt{m}w_0$、y 方向で $2\sqrt{n}w_0$ になります。ここでいうｍ、ｎとは以下で述べる、laser_mode_m と laser_mode_n のことです。

waist_position　：　最もビームが絞られた点（ウェスト）の空間座標を指定します。

beam_direction　：　ビームによって運搬されるエネルギーの方向を単位ベクトルで指定します。

up_direction　：　ビームが流れる方向と垂直な面内における y 方向を単位ベクトルで指定します。





laser_mode　:　レーザービームの種類を指定します。以下の４つを指定できます。

Gaussian

Hermite_Gaussian

Laguerre_Gaussian

file_input

Gaussian を指定すると Gaussian beam が選択されます。

Hermite_Gaussian を指定すると Hermite_Gaussian beam が選択されます。

Laguerre_Gaussian を指定すると Laguerre_Gaussian beam が選択されます。

file_input を指定すると、ビームの断面形状を laser_profile.txt で決めることができます。

Gaussian beam

$$\tilde{u}(x,y,z)=\left(\frac{2}{\pi}\right)^{\frac{1}{2}}\frac{\exp\left[-ikz+i\psi(z)\right]}{w(z)}\exp\left[-\frac{x^2+y^2}{w^2(z)}-ik\frac{x^2+y^2}{2R(z)}\right]$$

Hermite-Gaussian Mode

$$\tilde{u}_{mn}(x,y,z)=\left(\frac{2}{\pi}\right)^{\frac{1}{2}}\frac{\exp\left[i(m+n+1)\psi(z)\right]}{\sqrt{2^m m!\,w(z)}\sqrt{2^n n!\,w(z)}}H_m\left(\frac{\sqrt{2}x}{w(z)}\right)H_n\left(\frac{\sqrt{2}y}{w(z)}\right)\exp\left[-ikz-ik\frac{x^2+y^2}{2R(z)}\right]$$

$$\exp\left[-\frac{x^2+y^2}{w^2(z)}\right]$$

Laguerre-Gaussian Mode

$$\tilde{u}_{pm}(r,\theta,z)=\sqrt{\frac{2p!}{\pi(m+p)!}}\frac{\exp[i(2p+m+1)(\psi(z)-\psi_0)]}{w(z)}\left(\frac{\sqrt{2}r}{w(z)}\right)^m L_p^m\left(\frac{2r^2}{w^2(z)}\right)$$

$$\exp\left[-ikz-ik\frac{r^2}{2R(z)}-im\theta-\frac{r^2}{w^2(z)}\right]$$

ここで

$$w(z)=w_0\sqrt{1+\left(\frac{z}{z_R}\right)^2}$$

$$R(z)=z+\frac{{z_R}^2}{z}$$

$$\psi(z)=\tan^{-1}\left(\frac{z}{z_R}\right)$$

$$z_R=\frac{1}{2}k{w_0}^2=\frac{\pi}{\lambda}{w_0}^2$$

です。その他の詳細はチュートリアル４を参照してください。

laser_order_m　：　Hermite-Gaussian Mode の添え字 m、Laguerre-Gaussian Mode の添え字 m、を指定します。

laser_order_n　：　Hermite-Gaussian Mode の添え字 n、Laguerre-Gaussian Mode の添え字 p、を指定します。

times_diffraction_limited　：　現在、未使用です。
quality_factor　：　現在、未使用です。

include_radiation_heat_transfer　：　固液界面の熱輻射による冷却を計算に組み込むかどうかを指定するためのフラグです。熱輻射によるエネルギー損失は以下の式で与えられます。

$q_R=\sigma\varepsilon\left(T^4-T_\infty^4\right)$　　W/m^2

ここでσは Stefan-Boltzmann constant、εは emissivity、T_∞は ambient temperature です。

ambient_temperature　：　ambient temperature をケルヴィン単位で指定します。

TecPlot file
write_velocity　：　速度の計算結果を出力をするかしないかを指定するためのフラグです。.true.で出力します。.false.で出力しません。

write_pressure　：　pressure の計算結果を出力をするかしないかを指定するためのフラグです。
.true.で出力します。.false.で出力しません。

write_temperature　：　温度の計算結果を出力をするかしないかを指定するためのフラグです。.true.で出力します。.false.で出力しません。

write_signed_distance　：　signed distance 関数の計算結果を出力するかしないかを指定するためのフラグです。.true.で出力します。.false.で出力しません。

write_liquid_mass_fraction　：　liquid mass fraction の計算結果を出力するかしないかを指定するためのフラグです。.true.で出力します。.false.で出力しません。

write_solid_mass_fraction　：　solid mass fraction の計算結果を出力するかしないかを指定するためのフラグです。.true.で出力します。.false.で出力しません。

2.2 material.txt

air
gas_density　：　気相の密度を指定します。単位は、kg/m^3 です。

gas_specific_heat　：　気相の定圧比熱を指定します。単位は、$J/(kg \cdot K)$ です。

gas_viscosity　：　気相の粘性係数を指定します。単位は、$kg/(m \cdot \sec)$ です。

gas_thermal_conductivity　：　気相の熱伝導率を指定します。単位は、$W/(mK)$ です。

gas_volume_expansion　：　気相の体膨張率を指定します。単位は、$1/K$ です。

solid
solid_density　：　固相の密度を指定します。単位は、kg/m^3 です。

solid_specific_heat　：　固相の定圧比熱を指定します。単位は、$J/(kg \cdot K)$です。

solid_thermal_conductivity　：　固相の熱伝導率を指定します。単位は、$W/(mK)$です。

solid_volume_expansion　：　固相の体膨張率を指定します。単位は、$1/K$です。

liquid
liquid_density　：　液相の密度を指定します。単位は、kg/m^3です。

liquid_specific_heat　：　液相の定圧比熱を指定します。単位は、$J/(kg \cdot K)$です。

liquid_thermal_conductivity　：　液相の熱伝導率を指定します。単位は、$W/(mK)$です。

liquid_viscosity　：　液相の粘性係数を指定します。単位は、$kg/(m \cdot \sec)$です。

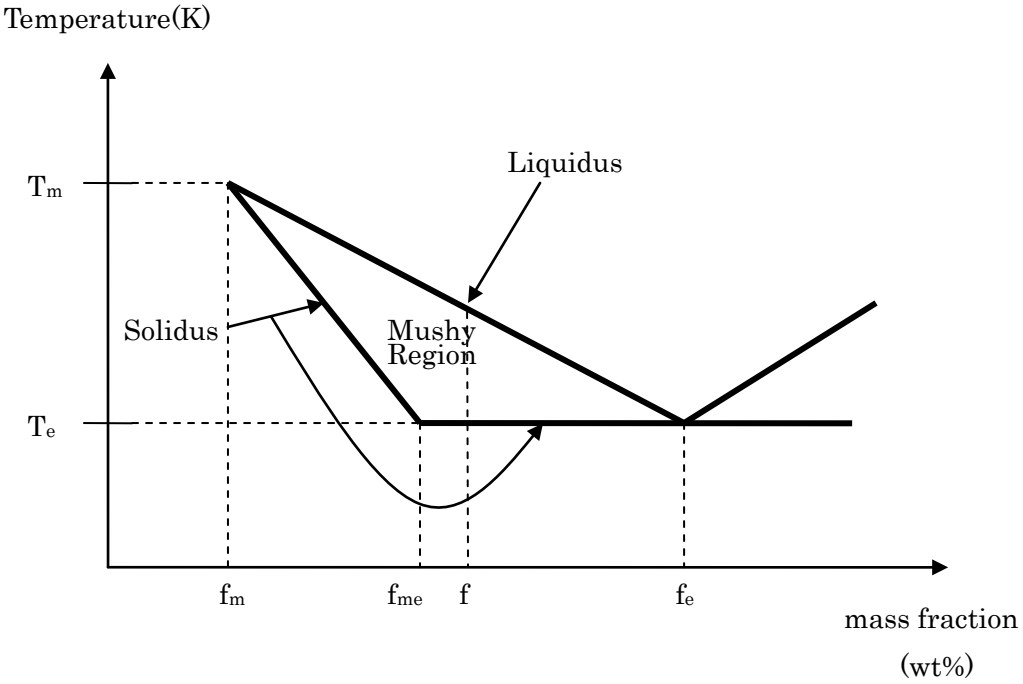
liquid_volume_expansion　：　液相の体膨張率を指定します。単位は、$1/K$です。

surface_tension　：　気液界面の表面張力σを指定します。単位は、N/mです。

dsurface_tension_dt　：　表面張力の温度微分$\frac{d\sigma}{dT}$を指定します。単位は、$N/(mK)$です。

thermodynamics

現在の実装では、解析的な取り扱いが容易になるように、Fe-C 系の phase-diagram の特徴をあらかじめ取り込んだ形になっています。ですが、いくつかの注意と近似のもとに、かなり一般の系に適用することができます。



まず最初の近似は、Solidus line と Liquidus line が直線で近似できるとするものです。多くの系では近似的にしか成り立ちませんが、こうすることで内部の処理が大幅に高速化します。

latent_heat_of_fusion　：　融解熱を指定します。単位は、J/kg です。

latent_heat_of_vaporization　：　気化熱を指定します。単位は、J/kg です。ただし、今回の実装では、vaporization については、考慮していません。

melting_temperature　：　融点を指定します。単位は、K です。図のT_mを指定してください。

normal_boiling_temperature　：　沸点を指定します。単位は、K です。ただし、今回の実装では、vaporization については、考慮していません。

critical_point_temperature　：　臨界点を指定します。単位は、K です。ただし、今回の

実装では、vaporization については、考慮していません。

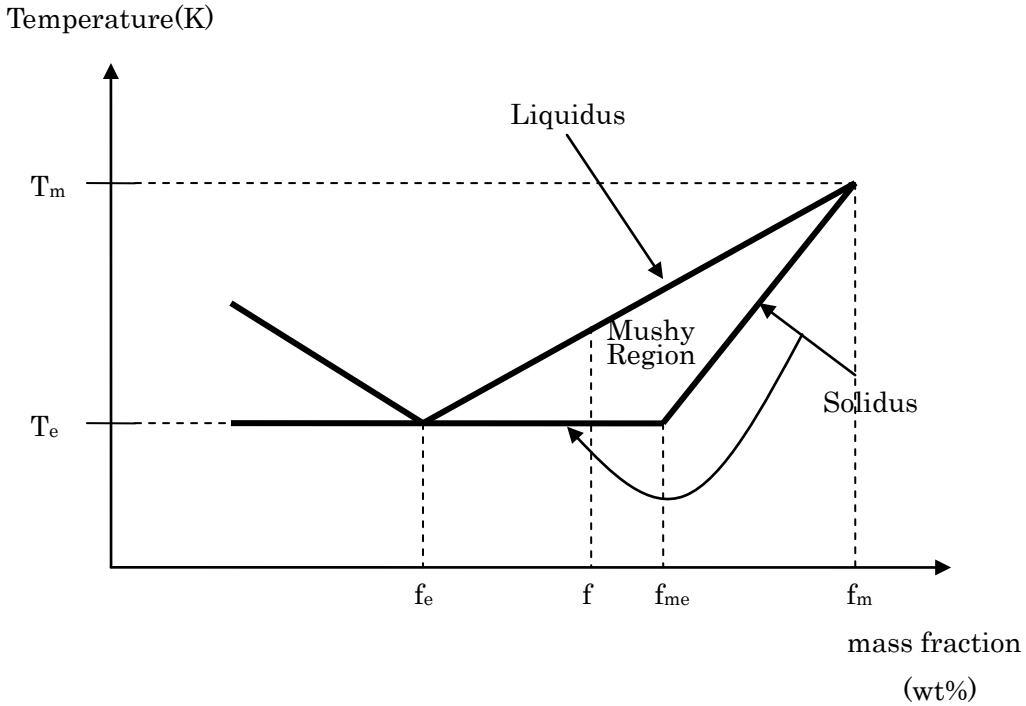
eutectic_temperature　：　共融点を指定します。単位は、K です。図のT_eを指定してください。

eutectic_concentration　：　図の $f_e - f_m$ を指定してください。単位は、下の concentration と同一にとってください。多くの場合、weight-%を用います。

equilibrium_partition_ratio　：　図の $\frac{f_{me} - f_m}{f_e - f_m}$ を指定してください。無次元量です。

concentration　：　今考えている系の concentration を f としたとき、図の $f - f_m$ を指定してください。単位は上の eutectic_concentration と同一にとってください。多くの場合、weight-%を用います。 f の値としては、f_m から f_e までを指定できます。

phase-diagram が以下のようにかかれる場合には、eutectic_concentration と concentration が正になるように上の記述を変更してください。すなわち、



eutectic_concentration　：　図の $f_m - f_e$ を指定してください。単位は、下の concentration と同一にとってください。多くの場合、weight-%を用います。

concentration　：　今考えている系の concentration を f としたとき、図の $f_m - f$ を指定してください。単位は上の eutectic_concentration と同一にとってください。多くの場合、weight-%を用います。 f の値としては、f_e から f_m までを指定できます。

laser_absorptivity　：　レーザーの absorptivity を指定します。無次元量です。

laser_emissivity　：　レーザーの emissivity を指定します。無次元量です。

permeability_coefficient　：　現在、未使用です。

primary_dendrite_arm_spacing　：　primary dendrite arm spacing を指定してください。単位はメートルです。Darcy force の permeability coefficient を以下の式から算出しています。

$$K_0 = 6.0 \times 10^{-4} \times \delta^2$$　　ここで δ は primary dendrite arm spacing です。

Permeability coefficient を指定したいときは、上の式を逆に解いて、

$$\delta = \sqrt{\frac{K_0}{6.0 \times 10^{-4}}}$$

から primary_dendrite_arm_spacing を指定してください。

2.3 reference.txt

reference_length　：　対象とする系の代表的な長さを指定してください。単位は m です。

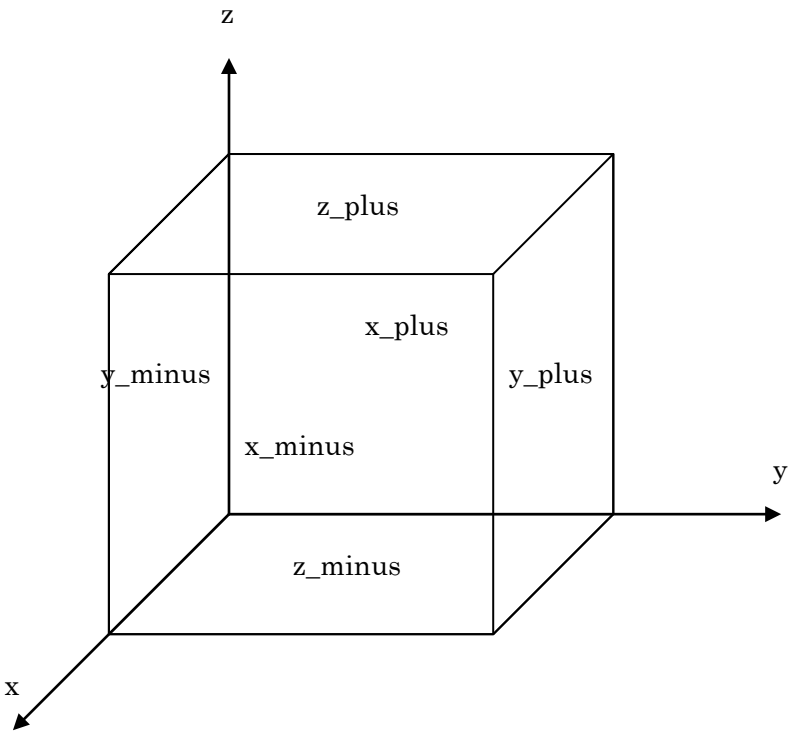
reference_velocity　：　対象とする系の代表的な速度を指定してください。単位は $m/\sec^2$ です。

base_temperature　：　対象とする系の代表的な温度を指定してください。単位は K です。

difference_temperature　：　対象とする系の代表的な温度の変化分を指定してください。単位は K です。

base_pressure　：　対象とする系の代表的な圧力を指定してください。単位は Pa です。

2.4 boundary.txt



上の図に示した表記で、計算領域の各境界面を、表します。つまり

x_minus： $x = 0$ 表面

x_plus ： $x = L = nx \times dx$ 表面

y_minus： $y = 0$ 表面

y_plus ： $y = M = ny \times dy$ 表面

z_minus： $z = 0$ 表面

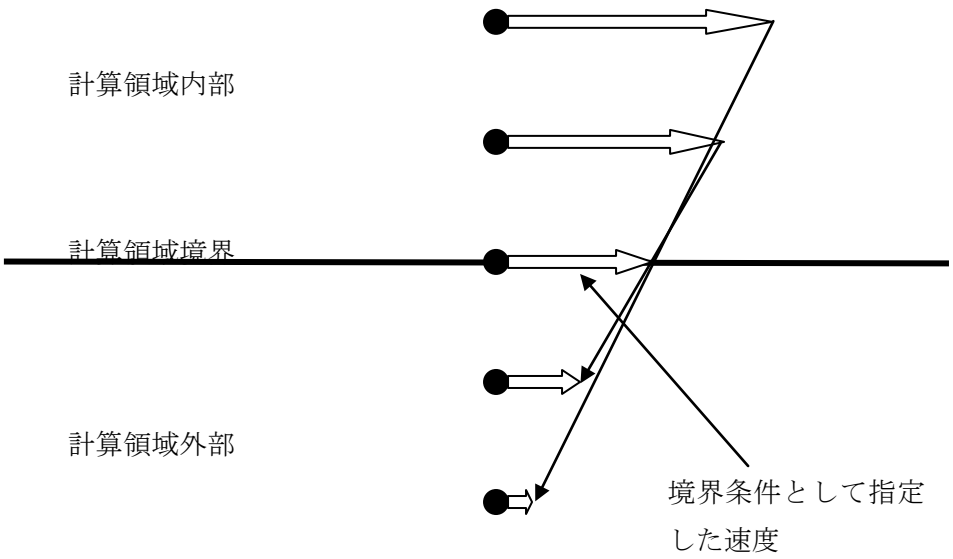
z_plus ： $z = N = nz \times dz$ 表面

速度の境界条件としては wall と dirichlet の２つを指定できます。

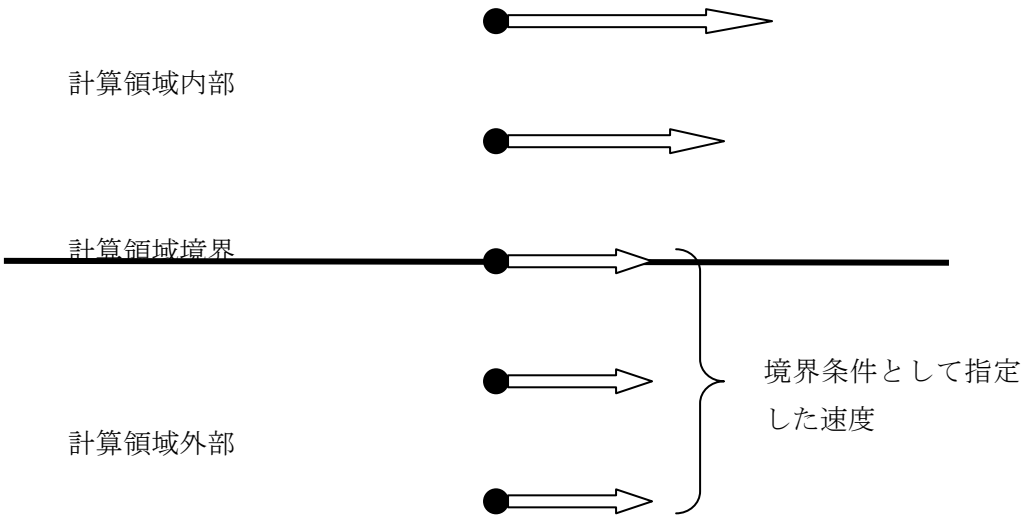
wall を指定すると下の図に示すように、境界上で指定した速度と内部領域で計算された速度を用いて、外部領域における速度が外挿により決められます。

dirichlet を指定すると、境界上、および外部領域における速度は、指定した速度で決められます。

速度の **wall** 境界条件



速度の **dirichlet** 境界条件



x_minus_velocity_bc　:　x_minus 境界面における速度の境界条件を指定します。wall

と dirichlet のふたつを指定できます。

x_minus_velocity　：　x_minus 境界面における速度を指定します。単位は $m/\sec$ です。

x_plus_velocity_bc　：　x_plus 境界面における速度の境界条件を指定します。wall と dirichlet のふたつを指定できます。

x_plus_velocity　：　x_plus 境界面における速度を指定します。単位は $m/\sec$ です。

y_minus_velocity_bc　：　y_minus 境界面における速度の境界条件を指定します。wall と dirichlet のふたつを指定できます。

y_minus_velocity　：　y_minus 境界面における速度を指定します。単位は $m/\sec$ です。

y_plus_velocity_bc　：　y_plus 境界面における速度の境界条件を指定します。wall と dirichlet のふたつを指定できます。

y_plus_velocity　：　y_plus 境界面における速度を指定します。単位は $m/\sec$ です。

z_minus_velocity_bc　：　z_minus 境界面における速度の境界条件を指定します。wall と dirichlet のふたつを指定できます。

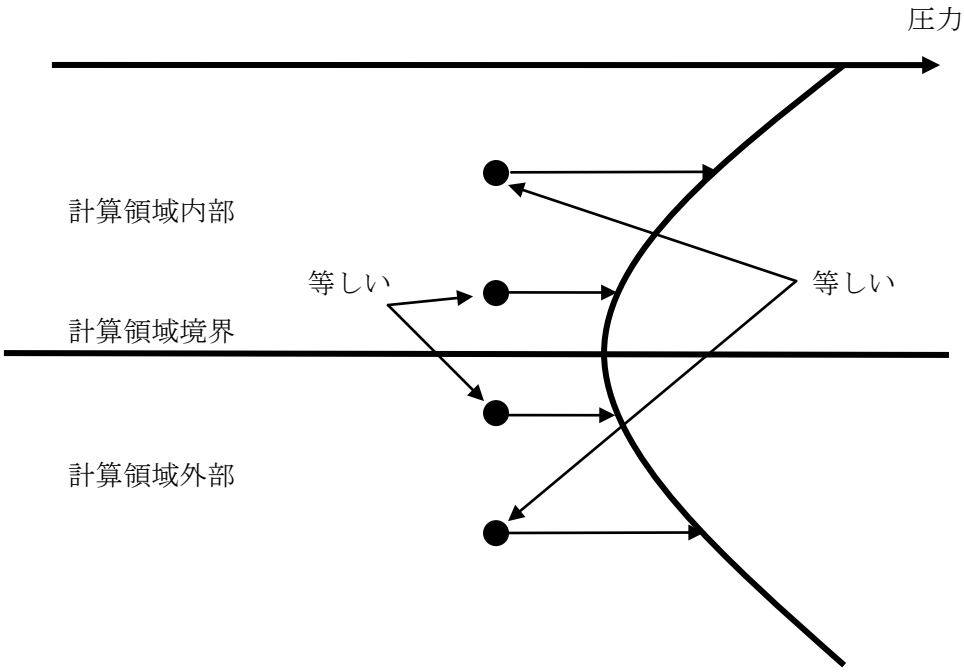
z_minus_velocity　：　z_minus 境界面における速度を指定します。単位は $m/\sec$ です。

z_plus_velocity_bc　：　z_plus 境界面における速度の境界条件を指定します。wall と dirichlet のふたつを指定できます。

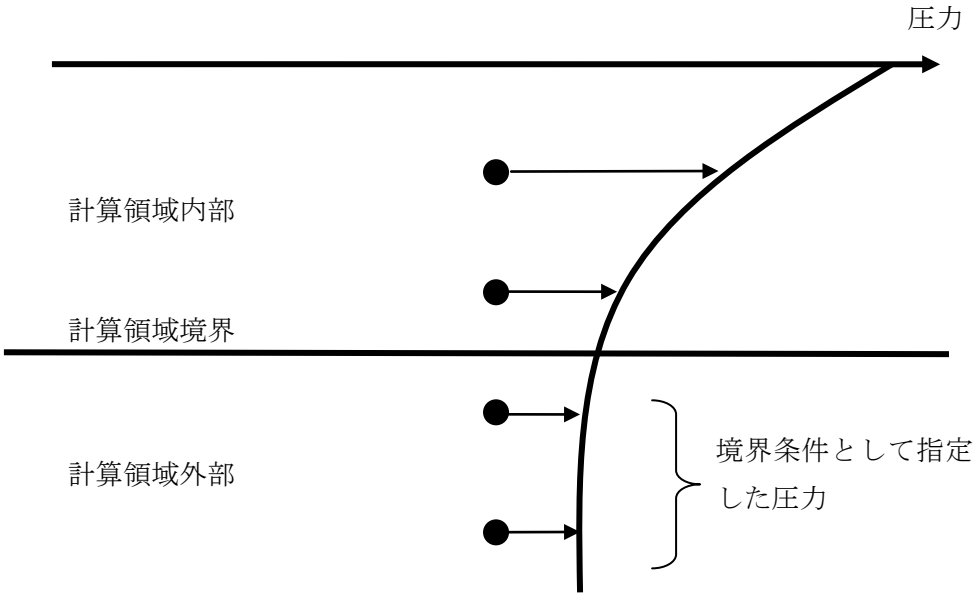
z_plus_velocity　：　z_plus 境界面における速度を指定します。単位は $m/\sec$ です。

圧力の境界条件としては、grad_zero と dirichlet の２つを指定できます。
gard_zero を指定すると、境界上で境界面に垂直な方向への圧力の交配が 0 になるように計算領域外部の圧力を設定します。
dirichlet を指定すると、指定した圧力で計算領域外部の圧力が設定されます。下の図を参照してください。

圧力の **grad_zero** 境界条件



圧力の **dirichlet** 境界条件



x_minus_pressure_bc　：　x_minus 境界面における圧力の境界条件を指定します。

grad_zero と dirichlet のふたつを指定できます。

x_minus_pressure　：　x_minus_pressure_bc が dirichlet のときに、x_minus 境界面における圧力を指定します。単位は *Pa* です。

x_plus_pressure_bc　：　x_plus 境界面における圧力の境界条件を指定します。
grad_zero と dirichlet のふたつを指定できます。

x_plus_pressure　：　x_plus_pressure_bc が dirichlet のときに、x_plus 境界面における圧力を指定します。単位は *Pa* です。

y_minus_pressure_bc　：　y_minus 境界面における圧力の境界条件を指定します。
grad_zero と dirichlet のふたつを指定できます。

y_minus_pressure　：　y_minus_pressure_bc が dirichlet のときに、y_minus 境界面における圧力を指定します。単位は *Pa* です。

y_plus_pressure_bc　：　y_plus 境界面における圧力の境界条件を指定します。
grad_zero と dirichlet のふたつを指定できます。

y_plus_pressure　：　y_plus_pressure_bc が dirichlet のときに、y_plus 境界面における圧力を指定します。単位は *Pa* です。

z_minus_pressure_bc　：　z_minus 境界面における圧力の境界条件を指定します。
grad_zero と dirichlet のふたつを指定できます。

z_minus_pressure　：　z_minus_pressure_bc が dirichlet のときに、z_minus 境界面における圧力を指定します。単位は *Pa* です。

z_plus_pressure_bc　：　z_plus 境界面における圧力の境界条件を指定します。
grad_zero と dirichlet のふたつを指定できます。

z_plus_pressure　：　z_plus_pressure_bc が dirichlet のときに、z_plus 境界面における圧力を指定します。単位は *Pa* です。

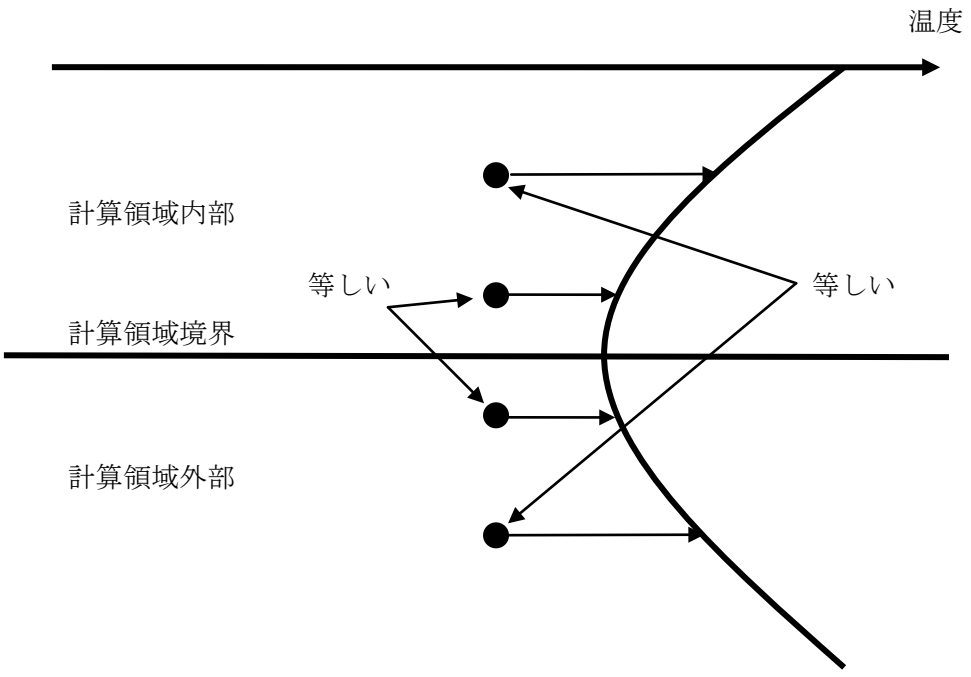
温度の境界条件としては、adiabatic と dirichlet と heat_transfer の 3 つを指定できます。

adiabatic を指定すると、境界上で境界面に垂直な方向への温度の交配が 0 になるように計算領域外部の温度を設定します。いわゆる断熱境界条件になります。
dirichlet を指定すると、指定した温度で計算領域外部の温度が設定されます。いわゆる定温境界条件になります。
heat_transfer を指定すると、境界における熱流速が以下の関係式を満たすように、計算領域外部の温度が設定されます。

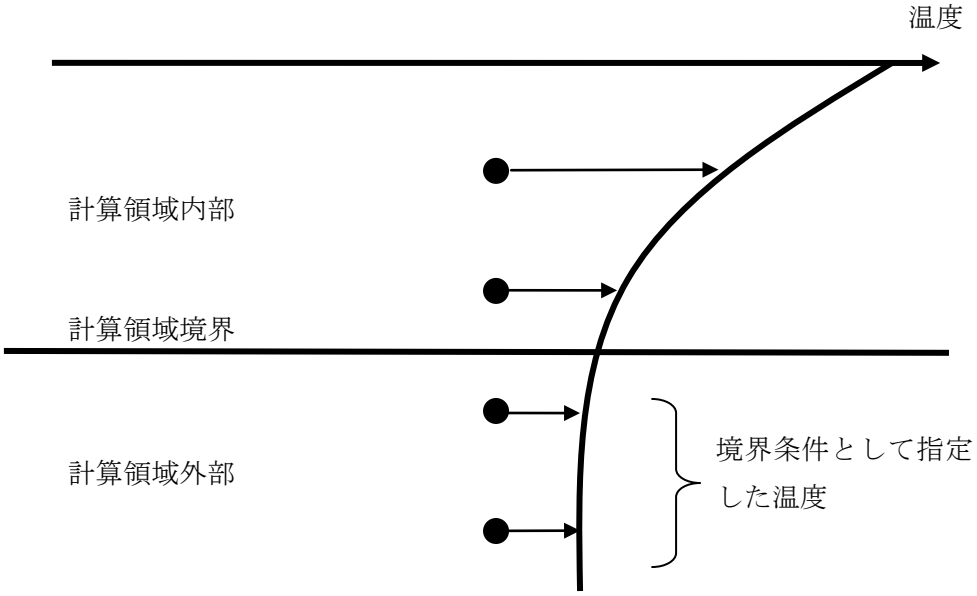
$$-k_B \left.\frac{\partial T}{\partial n}\right|_{Boundary} = h\left(T_{Boundary} - T_\infty\right)$$

ここで、k_B は当該境界面に接する、計算領域内部の熱伝導率です。$\frac{\partial}{\partial n}$ は境界に垂直に境界内部から外部に向けて取った、微分作用素です。$T_{Boundary}$ は境界面の温度、T_∞ は、系を取り巻く雰囲気の温度で、先述の ambient temperature の値を用いています。この境界条件は、Newton の冷却則が成立するような系において、そのときの気相の convection に由来する熱損失を、モデル化するものとして使用しています。h は heat transfer coefficient で、系の冷却曲線などから決めることのできる定数です。
それぞれ下の図を参照してください。

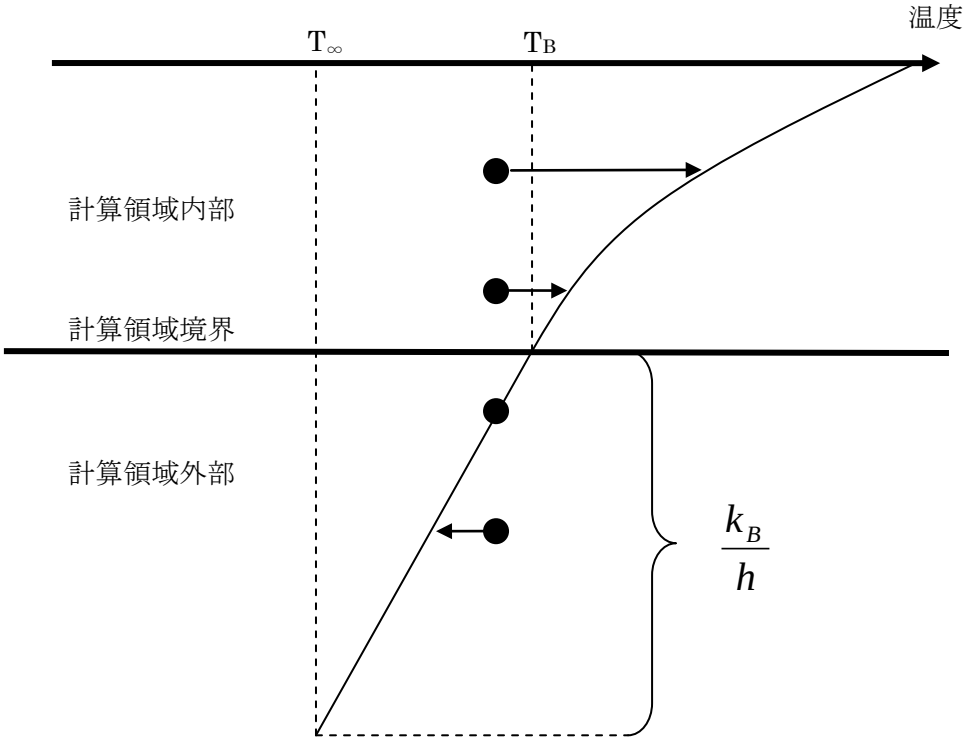
温度の **adiabatic** 境界条件



温度の **dirichlet** 境界条件



温度の **heat_transfer** 境界条件



x_minus_temperature_bc　：　x_minus 境界面における温度の境界条件を指定します。adiabatic と dirichlet と heat_transfer の３つを指定できます。

x_minus_temperature　：　x_minus_temperature_bc が dirichlet のときに、x_minus 境界面における温度を指定します。単位は *K* です。

x_plus_temperature_bc　：　x_plus 境界面における温度の境界条件を指定します。adiabatic と dirichlet と heat_transfer の３つを指定できます。

x_plus_temperature　：　x_plus_temperature_bc が dirichlet のときに、x_plus 境界面における温度を指定します。単位は *K* です。

y_minus_temperature_bc　：　y_minus 境界面における温度の境界条件を指定します。adiabatic と dirichlet と heat_transfer の３つを指定できます。

y_minus_temperature　：　y_minus_temperature_bc が dirichlet のときに、y_minus 境界面における温度を指定します。単位は *K* です。

y_plus_temperature_bc　：　y_plus 境界面における温度の境界条件を指定します。adiabatic と dirichlet と heat_transfer の３つを指定できます。

y_plus_temperature　：　y_plus_temperature_bc が dirichlet のときに、y_plus 境界面における温度を指定します。単位は *K* です。

z_minus_temperature_bc　：　z_minus 境界面における温度の境界条件を指定します。adiabatic と dirichlet と heat_transfer の３つを指定できます。

z_minus_temperature　：　z_minus_temperature_bc が dirichlet のときに、z_minus 境界面における温度を指定します。単位は *K* です。

z_plus_temperature_bc　：　z_plus 境界面における温度の境界条件を指定します。adiabatic と dirichlet と heat_transfer の３つを指定できます。

z_plus_temperature　：　z_plus_temperature_bc が dirichlet のときに、z_plus 境界面における温度を指定します。単位は *K* です。

heat_transfer_coefficient　：　heat transfer coefficient の値を設定します。単位は $W/\left(m^2 \cdot K\right)$ です。

temperature_extrapolation_order　：　現在、未使用です。

2.5　geometry.txt

geometry.txt についてはチュートリアル３で説明しています。そちらも参照してください。geometry.txt は signed distance 関数を初期化するために読み込まれるファイルです。計算初期における、気相とその他の相（固相と液相）の界面を決めます。input.txt 内で
sdf_initial_mode　=　file_input
と指定すると、ソルバーから読み込まれます。
フォーマットはチュートリアル３および、input.txt の sdf_initial_mode の説明の項で述べてありますので、そちらを参照してください。

2.6　laser_schedule.txt

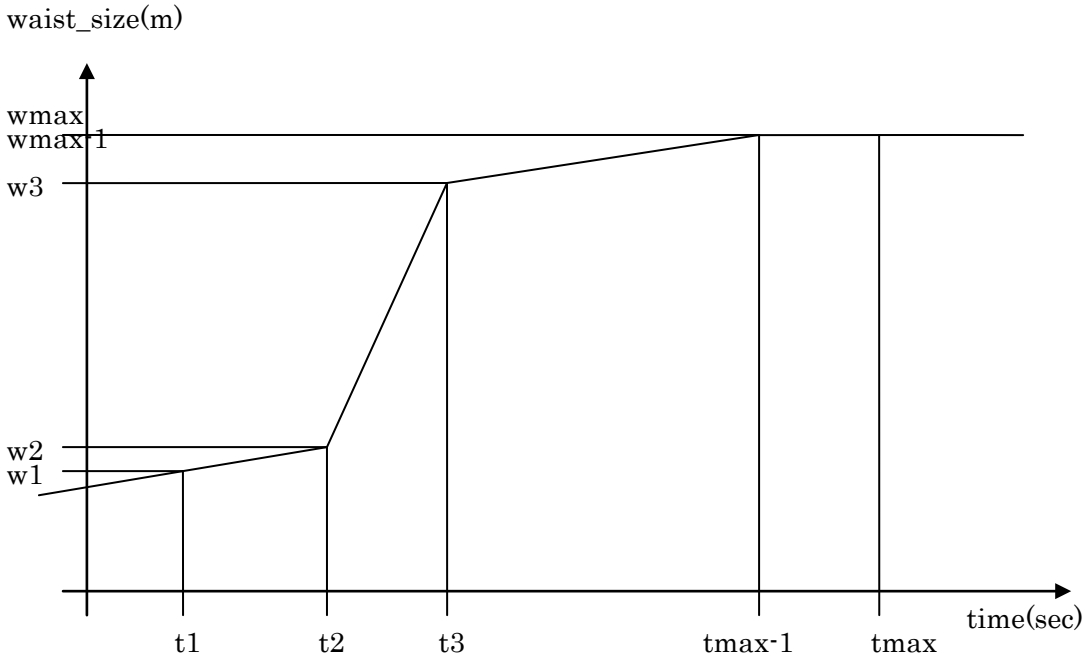
laser_schedule.txt についてはチュートリアル５で説明しています。そちらも参照してください。laser_schedule.txt はレーザービームの各種パラメータを時間的に変化させるときに用います。レーザービームの各種パラメータについてはチュートリアル４、ならびに、input.txt の以下の項目を参照してください。

Laser beam parameters
include_laser_beam
use_laser_schedule
laser_wave_length
power_transmitted_by_the_beam
waist_size
waist_position
beam_direction
up_direction
laser_mode
laser_order_m
laser_order_n

laser_schedule.txt は input.txt の use_laser_schedule を.true.にすると本体のプログラム

から読み込まれます。フォーマットは以下の通りです。

number_of_data
time1(sec)　power1(Wat)　waist_size1(m)　waist_position1(m)(x,y,z)…….
time2(sec)　power2(Wat)　waist_size2(m)　waist_position2(m)(x,y,z) …….
time3(sec)　power3(Wat)　waist_size3(m)　waist_position3(m)(x,y,z) …….
.....
waist_size を例にとると



時刻 t1 以前の値は、時刻 t1 の値 w1 と時刻 t2 の値 w2 を線形に外挿して決めます。

時刻 t　$t_i \le t < t_j$　の値は時刻 ti の値 wi と時刻 tj の値 wj を線形内挿して決めます。

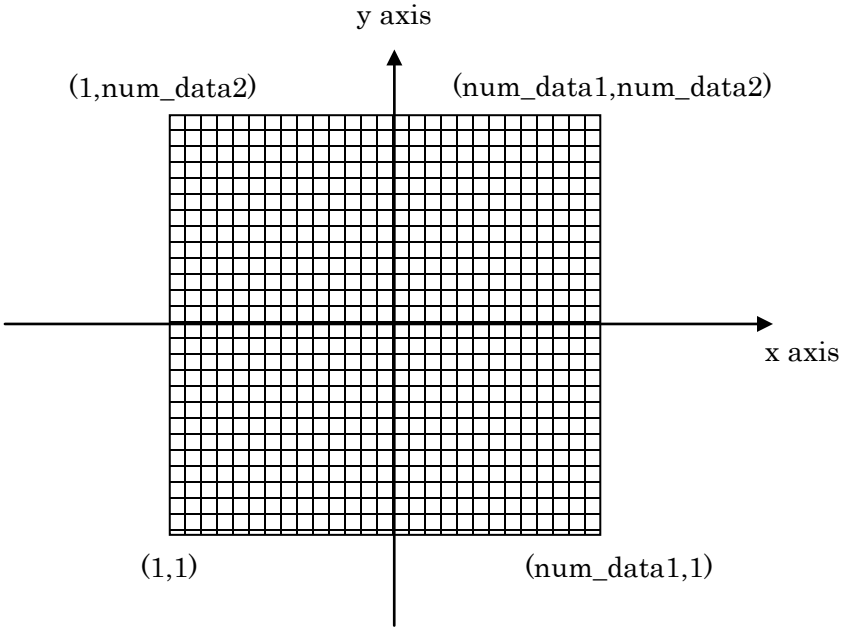
時刻 $t_{\max}$ 以降の値は時刻 $t_{\max-1}$ の値 $w_{\max-1}$ と時刻 $t_{\max}$ の値 $w_{\max}$ を線形に外挿して決めます。
より詳細は、チュートリアル 5 を参照してください。

2.7　laser_profile.txt

laser_profile.txt についてはチュートリアル 4 で説明しています。そちらも参照してください。laser_profile.txt はレーザービームに垂直な断面内におけるレーザービームの強度をユーザが自分で指定したいときに用います。input.txt の laser_mode を file_input に指定すると、プログラム本体から読み込まれます。
laser_profile.txt のフォーマットは以下のようになります。

num_data1　　num_data2
data(1,1)
data(2,1)
data(3,1)

data(num_data1,num_data2)



ここで num_data1 と num_data2 は必ずしも等しくとる必要はありませんが、どちらも奇数を指定してください。data(i,j)には図に示した、対応する点のレーザ強度を記述してください。強度は、相対的なものでいいです。内部で適切に規格化を行います。
より詳細はチュートリアル４を参照してください。