



Gmsh / Gnuplot / OpenFOAM / ParaView による
Open Source CAE Suite for Windows
建築分野向けチュートリアル

大嶋 拓也

新潟大学工学部

D R A F T

2008年4月18日版

分割版〔第1章〕

目次

1 基礎事項	3	2.5.8 解析の実行	24
1.1 数値シミュレーションとは	3	2.5.9 結果の表示	24
1.2 微分とは	3	2.5.10 流線を描く	27
1.3 微分方程式とは	4	2.6 速度成分のプロット	28
1.4 解析解の求め方	4	2.6.1 sample ユーティリティの実行	28
1.4.1 一般解	4	2.6.2 Gnuplot による抽出したデータのプロット	31
1.4.2 初期条件・特殊解	4	2.7 メッシュの分割数を上げる（コマンドラインによる操作）	32
1.5 数値解の求め方	4	2.7.1 ケースのコピー	32
1.5.1 離散化	4	2.7.2 blockMeshDict ファイルの編集とメッシュの生成	34
1.5.2 差分方程式の解法	5	2.7.3 初期値と境界条件の確認	34
1.5.3 離散化誤差	6	2.7.4 物性値の確認	35
1.6 偏微分方程式・メッシュ・セル・境界条件	8	2.7.5 実行条件の制御	36
1.6.1 偏微分方程式	8	2.7.6 メッシュの確認	37
1.6.2 メッシュ	9	2.7.7 解析の実行	37
1.6.3 境界条件	9	2.7.8 結果の確認	37
1.7 シミュレーションの手順	10	2.7.9 速度成分のプロット	38
1.7.1 プリプロセッシング	10	3 高層ビル周り気流の解析	41
1.7.2 解析実施	10	3.1 概要	41
1.7.3 ポストプロセッシング	11	3.1.1 本章解析ケースの紹介	41
1.8 本チュートリアルで使用するソフトウェア	11	3.1.2 解析ケース概要	42
1.8.1 Gmsh	11	3.2 メッシュの作成	42
1.8.2 OpenFOAM	12	3.2.1 等間隔メッシュの作成	44
1.8.3 ParaView	12	3.2.2 メッシュの再分割	46
1.8.4 Gnuplot	12	3.2.3 メッシュの再々分割	49
2 キャビティ流れの解析	15	3.2.4 建物内のメッシュの除去	50
2.1 概要	15	3.2.5 高層建物周辺のメッシュ再分割	51
2.2 CFD とは	15	3.2.6 高層建物内のセルを取り除く	54
2.3 支配方程式	16	3.2.7 低層建物周辺メッシュの再分割	55
2.4 解析対象	16	3.3 初期値の設定・実行条件の設定	57
2.5 解析の実行手順	17	3.4 ソルバの実行	58
2.5.1 FoamX の起動	17	3.5 実行結果の確認	58
2.5.2 メッシュの作成	18	3.5.1 ベクトルプロットの作成	59
2.5.3 境界条件・初期条件の作成	21	3.5.2 流線のプロット	61
2.5.4 動粘性係数の設定	22	3.6 風洞実験結果との比較	63
2.5.5 実行条件の設定	22		
2.5.6 データの保存	23		
2.5.7 メッシュの確認	23		

1

基礎事項

1.1 数値シミュレーションとは

世の中の様々な物理現象の多くは、数学的には微分方程式によって記述される。その微分方程式の数値解をコンピュータを使って求めることを、数値シミュレーションと呼んでいる。

1.2 微分とは

ここでは質点の運動を例にとって、微分の復習から始めよう。

ある点 P が直線上を運動する場合を考えると、運動している物体の速度は、移動距離 ÷ 経過時間で与えられる。すなわち、物体の位置 x が時刻 t の関数であるとする、ある時刻 t_0 から Δt の間に $x(t_0)$ から $x(t_0 + \Delta t)$ まで移動したときの平均的な速度 U は

$$U = \frac{x(t_0 + \Delta t) - x(t_0)}{\Delta t} \quad (1.1)$$

である (図 1.1)。この平均的な速度を一般化していえば、 $x(t)$ の平均変化率である。

さらに、点 P の運動は急には変化しない、すなわち滑らかであると仮定して、 Δt を 0 へ近づけた極限を考える。このとき点 P の移動量 $x(t_0 + \Delta t) - x(t_0)$ も小さくなって、上式は Δt によらないある一定値に近づくと期待できる。これを数式で表すと

$$\begin{aligned} U(t_0) &= \lim_{\Delta t \rightarrow 0} \frac{x(t_0 + \Delta t) - x(t_0)}{\Delta t} \\ &= x'(t_0) \end{aligned} \quad (1.2)$$

となる。この $x'(t_0)$ を $x(t)$ の $t = t_0$ における微分係数といい、上式は時刻 t_0 における速度が位置の微分係数で与えられることを表している。

さらに、上式の t_0 は任意の時刻に置き換えられるから、速度 U を t の関数と考えると

$$\begin{aligned} U(t) &= \lim_{\Delta t \rightarrow 0} \frac{x(t + \Delta t) - x(t)}{\Delta t} \\ &= x'(t) \\ &= \frac{dx(t)}{dt} \end{aligned} \quad (1.3)$$

と書くことができる。これら $x'(t)$ および $dx(t)/dt$ を $x(t)$ の導関数または微分といい、上式は任意

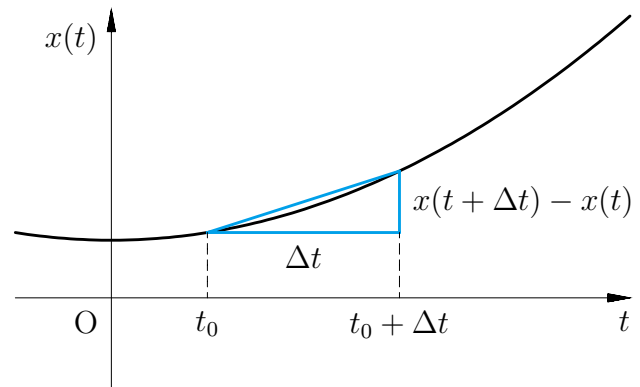


図 1.1 平均変化率の概念

の時刻における速度が位置の微分で与えられることを表している。この関係は、 $x(t)$ がどのような形の未知の関数であっても、滑らか、すなわち微分可能であれば成り立つことに注意してほしい。

1.3 微分方程式とは

前節では物体の位置の微分が速度であることを示したが、このように「微分」という概念を用いることで、「位置」と「速度」などの物理的な量を表す関数を、それらの形が未知のままであっても、微分を介して一般的な形で結びつける、すなわち等式として表すことができる。未知の関数の微分を含む等式を微分方程式という。

例えば、速度が時間によらず一定値 U_0 である場合、 $x(t)$ は以下の微分方程式を満たす。

$$\frac{dx(t)}{dt} = U_0 \quad (1.4)$$

微分方程式を満たす関数は、その微分方程式の解であるといい、全ての解を求めることを、その微分方程式を解くという。この微分方程式を解く方法としては、大きく分けて2種類が挙げられる。1つは微分方程式を満たす関数の式を求める方法であり、もう一つは微分方程式を満たす関数の値を求める方法である。前者によって求められた解を解析解、後者による解を数値解と呼ぶ。

1.4 解析解の求め方

1.4.1 一般解

式 (1.4) の微分方程式における微分の項、すなわち左辺は未知関数 $x(t)$ の t による微分であるから、 $x(t)$ を求めるには式 (1.4) の両辺を t で積分すればよい。そのようにすると、

$$x(t) = U_0 t + C \quad (1.5)$$

が得られる。ただし、 C は任意の定数である。微分方程式を解くと一般に、任意の定数を含んだ解が得られる。この解の形式を一般解という。

1.4.2 初期条件・特殊解

式 (1.5) の定数 C を定めるには、さらに条件が必要である。一般に数値シミュレーションでは、初期時刻 $t = 0$ における x の値、すなわち $x(0)$ をあらかじめ決めて与えることで、 C の値が定まるようにする。この時刻 $t = 0$ において与える条件のことを初期条件という。

例えば式 (1.5) の場合、 $t = 0$ において $x(0) = 1$ と定めれば、 $C = 1$ と定まる。よって式 (1.5) は

$$x(t) = U_0 t + 1 \quad (1.6)$$

となる。このように、一意に定まった解の形式を特殊解という。

1.5 数値解の求め方

1.5.1 離散化

数値解の求め方の全ての出発点は、式 (1.3) の微分の定義式において Δt が無限に 0 に近いと

考えたのに対し、式 (1.1) のように、ある有限の大きさを持つと考えることにある。すなわち、微分 $dx(t)/dt$ を、微分の定義式から \lim を外した形の

$$\frac{dx(t)}{dt} \approx \frac{x(t + \Delta t) - x(t)}{\Delta t} \quad (1.7)$$

または、 t 軸方向の対称性を考慮して

$$\frac{dx(t)}{dt} \approx \frac{x(t + \Delta t/2) - x(t - \Delta t/2)}{\Delta t} \quad (1.8)$$

によって近似することにある (\approx は近似を表す)。これらの式の形を総称して差分といい、式 (1.7) を特に前進差分、式 (1.8) を中心差分と呼んでいる。さらに、微分方程式の微分を差分に置き換えることを離散化、離散化された微分方程式を差分方程式という。

離散化は 2 階以上の微分に対しても可能であり、例えば $x(t)$ の 2 階微分に対して式 (1.8) の中心差分を 2 回適用すると、以下の離散化式が得られる。

$$\begin{aligned} \frac{d^2x(t)}{dt^2} &= \frac{d}{dt} \frac{dx(t)}{dt} \approx \frac{x'(t + \Delta t/2) - x'(t - \Delta t/2)}{\Delta t} \\ &\approx \frac{\frac{x(t + \Delta t) - x(t)}{\Delta t} - \frac{x(t) - x(t - \Delta t)}{\Delta t}}{\Delta t} \\ &= \frac{x(t + \Delta t) - 2x(t) + x(t - \Delta t)}{\Delta t^2} \end{aligned} \quad (1.10)$$

式 (1.4) を前進差分によって離散化すると、以下となる。

$$\frac{x(t + \Delta t) - x(t)}{\Delta t} = U_0 \quad (1.9)$$

1.5.2 差分方程式の解法

式 (1.9) の両辺に Δt をかけ、整理すると

$$x(t + \Delta t) = x(t) + U_0 \Delta t \quad (1.11)$$

のように、初期条件 $x(0)$ が与えられると Δt だけ後の $x(t)$ の値が順次算出される、一種の漸化式となる。ここで U_0 は問題の条件から与えられ、 Δt は解析者が任意に決定する。 Δt の決め方にはある程度の指針があり、2.5.5 節で詳述する。この Δt を、時間刻みという。

ここでは例えば、 $U_0 = 2$ 、 $\Delta t = 0.1$ であるとすると、式 (1.11) は

$$x(t + 0.1) = x(t) + 0.2 \quad (1.12)$$

となり、さらに 1.4.2 節で特殊解を求めた場合と同様、 $t = 0$ において $x(0) = 1$ の初期条件を与えると

$$\begin{aligned} x(0) &= 1 \\ x(0.1) &= x(0) + 0.2 = 1 + 0.2 = 1.2 \\ x(0.2) &= x(0.1) + 0.2 = 1.4 \\ x(0.3) &= x(0.2) + 0.2 = 1.6 \\ &\dots \end{aligned} \quad (1.13)$$

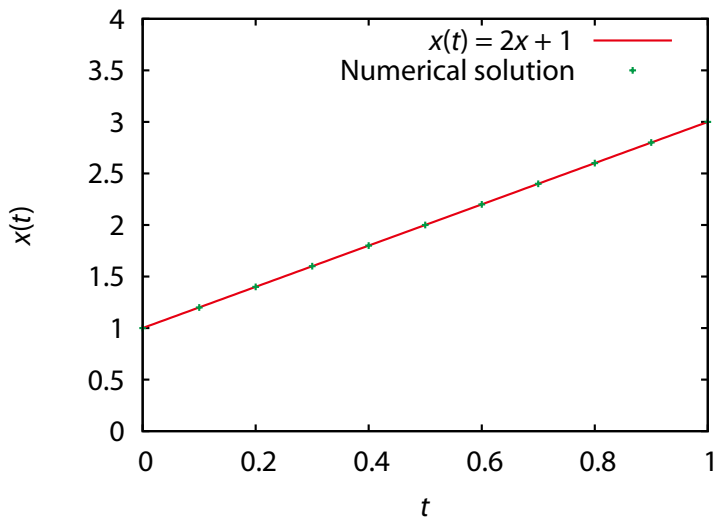


図 1.2 解析解と数値解の比較

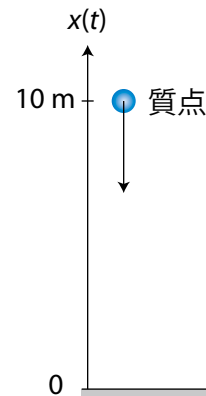


図 1.3 自由落下する質点

のように、 $x(t)$ の値が $t = 0$ から 0.1、すなわち Δt ごとに数値解が順次算出される。これが数値解の求め方の基本である。

さらに 1.4.2 節で得られた特殊解である式 (1.6) に $U_0 = 2$ を与えた

$$x(t) = 2t + 1 \quad (1.14)$$

と式 (1.13) の数値解を重ねてグラフにプロットすると、図 1.2 のようになる。この場合、解析解と数値解が一致していることがわかる。

解析解は厳密である一方、実際的な問題で解くべき微分方程式は、例えば第 2 章の式 (2.2)、式 (2.3) に示すように通常、式 (1.4) より遥かに複雑である。したがって、ほとんどの場合において、式 (1.5) のような解析解を求めることができない。一方、式 (1.7)、式 (1.8) に示したような差分化を微分方程式の各項に対してシステマティックに適用することで、解析解を求められない多くの複雑な微分方程式を数値解として解くことができる。このため実際的な問題では必ず、数値解を求めることとなる。本章冒頭で述べたとおり、数値シミュレーションとは通例、コンピュータを使って数値解を計算することを指す。

1.5.3 離散化誤差

それでは、解析解と数値解は必ず一致するか？

今度は図 1.3 のように、質点が重力下で自由落下する問題を考えてみよう。高さ方向に x 軸を取り、初期時刻 $t = 0$ において高さ 10 m の点で静止状態にあるとする。この自由落下を表す微分方程式は、重力加速度を 9.8 m/s^2 とすると、時刻 0 において速度 0 の等加速度運動であるから、以下となる。

$$\frac{dx(t)}{dt} = -9.8t \quad (1.15)$$

まず、この微分方程式の解析解を求めよう。一般解は、両辺を積分して

$$x(t) = -4.9t^2 + C \quad (1.16)$$

ただし、 C は積分定数である。さらに、 $x(0) = 10$ [m] の初期条件から $C = 10$ [m] が求められ、特殊解は以下のよう
に決まる。

$$x(t) = -4.9t^2 + 10 \quad (1.17)$$

つぎに数値解については、まず式 (1.14) 左辺の微分を前進差分で離散化すると、以下の差分方程式が得られる。

$$\frac{x(t + \Delta t) - x(t)}{\Delta t} = -9.8t \quad (1.18)$$

上式を整理すると、以下の漸化式が得られる。

$$x(t + \Delta t) = x(t) - 9.8t\Delta t \quad (1.19)$$

初期条件 $x(0) = 10$ [m] から始めて、 $\Delta t = 0.1$ [s] として 1.5.2 節と同様に上式を漸化式的に計算すると、次の結果が得られる。

$$\begin{aligned} x(0) &= 10 \\ x(0.1) &= 10 - 9.8 \times 0 \times 0.1 = 10 \\ x(0.2) &= 10 - 9.8 \times 0.1 \times 0.1 = 10 - 0.098 = 9.902 \\ x(0.3) &= 9.902 - 9.8 \times 0.2 \times 0.1 = 9.706 \\ x(0.4) &= 9.706 - 9.8 \times 0.3 \times 0.1 = 9.412 \\ &\dots \end{aligned} \quad (1.20)$$

式 (1.17) の解析解と式 (1.20) の数値解をグラフにプロットすると、図 1.4 のようになる。今度は図 1.2 と異なり、時間の経過に伴って数値解のプロットが解析解の曲線から外れており、解析解と数値解の間で差異が生じていることがわかる。このような数値解特有の誤差を、離散化誤差と呼んでいる。離散化誤差を抑制するには、どのようにすれば良いのだろうか。

次は時間刻み Δt の値を 4 分の 1 に小さくして、 $\Delta t = 0.025$ [s] として式 (1.18) を計算してみよう。

$$\begin{aligned} x(0) &= 10 \\ x(0.025) &= 10 - 9.8 \times 0 \times 0.025 = 10 \\ x(0.05) &= 10 - 9.8 \times 0.025 \times 0.025 = 9.994 \\ x(0.075) &= 9.994 - 9.8 \times 0.05 \times 0.025 = 9.982 \\ x(0.1) &= 9.982 - 9.8 \times 0.075 \times 0.025 = 9.963 \\ &\dots \end{aligned} \quad (1.20)$$

この計算結果を解析解と共にグラフにプロットすると、図 1.5 のようになる。図 1.4 の $\Delta t = 0.1$ [s] の場合と比べると、解析解と数値解の差が小さい、すなわち離散化誤差が小さくなっていることがわかる。一般に離散化における刻みを小さくすると、離散化誤差が小さくなる。ただし刻みを小さくすれ

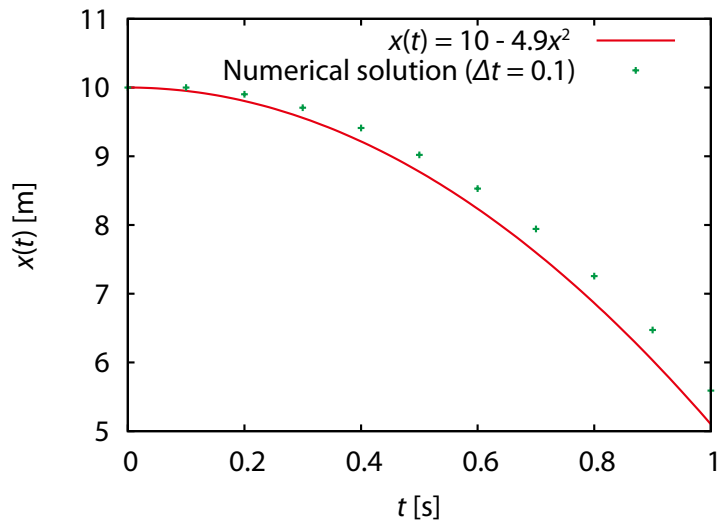


図 1.4 解析解と数値解の比較（時間刻み 0.1 秒）

ばその分、同じ時間幅を計算するために必要な時刻上の点数は増え、この場合は4倍となる。その分、計算量も増え、数値シミュレーションにおいてはより多くの所要時間を必要とすることとなる。

1.6 偏微分方程式・メッシュ・セル・境界条件

ここまで時間の離散化について説明してきたが、実際に我々がシミュレーションしようとする対象は、例えば、ビル周りの風、室内における音響伝搬のように、空間的にある程度の広がりを持っている。このシミュレーションすべき対象の空間を解析領域という。解析領域は、どのように離散化されるのだろうか。

1.6.1 偏微分方程式

「空間内のある点 x における時刻 $t + \Delta t$ での物理量 T が、時刻 t において点 x から Δx だけ離れた周囲の点における物理量の平均値として与えられる」という現象を、一次元で考えてみよう (図 1.6)。この現象は拡散と呼ばれ、建築における壁材料中の熱伝導、室内へのガスの放散等、身近な物理現象の多くにかかわっている。この現象を式で表すと、次のようになる。

$$T(x, t + \Delta t) = \frac{1}{2} \{T(x - \Delta x, t) + T(x + \Delta x, t)\} \quad (1.21)$$

上式を形式的に変形して、微分方程式を導こう。まず両辺から $T(x, t)$ を引くと

$$T(x, t + \Delta t) - T(x, t) = \frac{1}{2} \{T(x - \Delta x, t) - 2T(x, t) + T(x + \Delta x, t)\} \quad (1.22)$$

となり、両辺を Δt と Δx^2 で割ると

$$\frac{1}{\Delta x^2} \frac{T(x, t + \Delta t) - T(x, t)}{\Delta t} = \frac{1}{2\Delta t} \frac{T(x - \Delta x, t) - 2T(x, t) + T(x + \Delta x, t)}{\Delta x^2} \quad (1.23)$$

さらに整理して

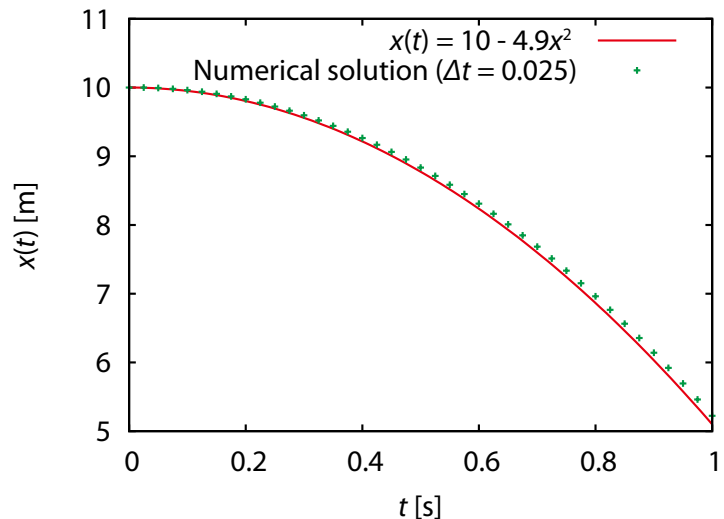


図 1.5 解析解と数値解の比較 (時間刻み 0.025 秒)

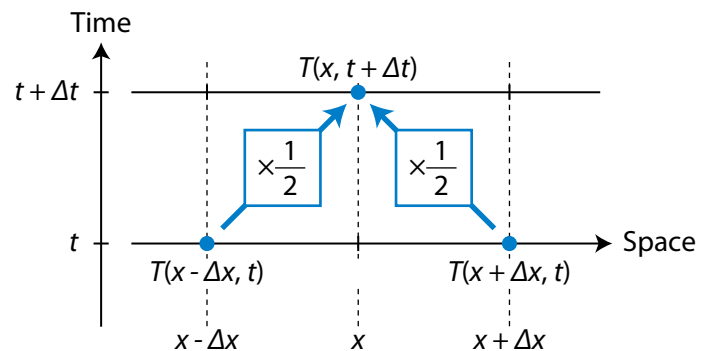


図 1.6 一次元における拡散のモデル

$$\frac{T(x, t + \Delta t) - T(x, t)}{\Delta t} = \frac{\Delta x^2}{2\Delta t} \frac{T(x - \Delta x, t) - 2T(x, t) + T(x + \Delta x, t)}{\Delta x^2} \quad (1.24)$$

となる。上式と式 (1.7) および式 (1.10) を比較すると、左辺は時刻 t に関する前進差分、右辺は位置 x に関する 2 階中心差分となっており、上式全体が差分方程式となっていることがわかる。さらに簡単のため右辺の $\Delta x^2/(2\Delta t)$ が 1 となるような関係を保ちながら、上式における Δt と Δx を 0 に近づけていくと、上式の差分が微分に置き換えられて、以下の拡散方程式と呼ばれる微分方程式となる。

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} \quad (1.25)$$

上式における微分は、被微分関数が時刻 t 、位置 x の 2 つの独立変数を含むために偏微分となり、したがって上式は偏微分方程式と呼ばれる。

1.6.2 メッシュ

式 (1.21) と式 (1.24) は形式的には同値であるから、式 (1.25) の偏微分方程式の数値解を求めるには式 (1.21) を計算すれば良い訳であるが、そのためには 1.5 節で時間軸を Δt ごとに区切ったように、空間方向にも Δx ごとに区切って、その区切られた各点ごとに物理量を計算すると都合が良い。すなわちそのように区切ることで、ある時刻における全ての区切られた点の値を式 (1.21) によって計算し、次に時刻を Δt だけ進めて、また同様に全ての点の値を式 (1.21) によって計算することで、数値解を順次求めることができる。

この区切りを 2 次元・3 次元空間に対して行くと、区切りが網の目のように見えることから、メッシュ、グリッド、格子などと呼ばれる。また、区切られたそれぞれのマス目のことをセルという。

1.6.3 境界条件

微分方程式を時間方向に解くときに、最初の時刻において初期条件が必要であったように、空間方向にも、解析領域の境界においては条件が必要である。これを境界条件という。

境界条件の主なものとしては、境界上における関数の値を定める種類の条件、および境界における関数の法線方向微分値を定める種類の条件が挙げられる。前者をディリクレ (Dirichlet) 型境界条件、後者をノイマン (Neumann) 型境界条件という。

数値解の計算においては、図 1.7 に示すように、境界上で境界外の値を参照することはできないため、式 (1.21) のような数値解の計算式そのものを使用することができない。したがって境界上の値は、境界条件から計算することになる。 $x = L$ が解析領域の境界、 $x < L$ が解析領域内部であるとすると、式 (1.21) の計

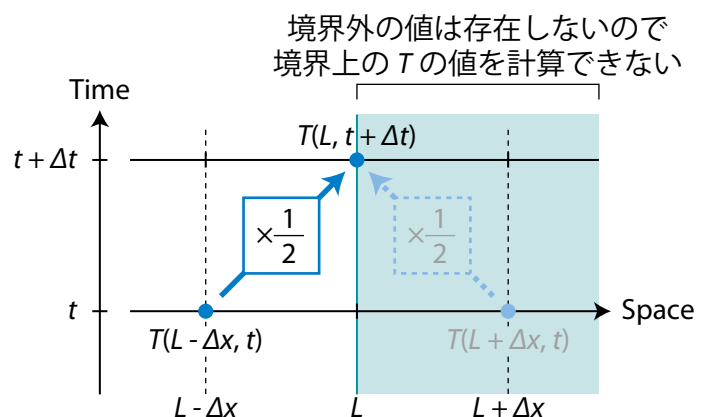


図 1.7 境界上の数値解

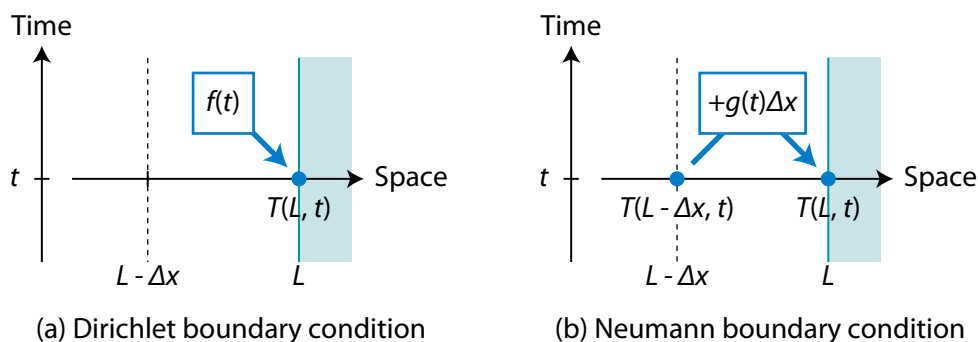


図 1.8 境界条件の取扱い

算においてディリクレ型境界条件を課す場合は、境界値を与える関数 $f(t)$ によって

$$T(L, t) = f(t) \quad (1.26)$$

として境界上の T の値を定める。またノイマン型境界条件の場合は、境界上の法線方向微分値を与える関数 $g(t)$ によって

$$T(L, t) = T(L - \Delta x) + g(t)\Delta x \quad (1.27)$$

のように境界値を定める。これらを図示すると図 1.8 のようになる。

1.7 シミュレーションの手順

以上が数値シミュレーションのごく簡単な原理である。それでは、この数値シミュレーションを具体的にどのように実施するのだろうか。

数値シミュレーションを実施することを、単に解析と呼ぶことが多い。本チュートリアルでも、以後そのように呼ぶこととする。解析の大まかな手順は、図 1.9 のようになる。

1.7.1 プリプロセッシング

解析のためには、まず解析対象の形状を何らかの方法によってコンピュータに入力して、メッシュデータを作成する必要がある。ついで境界条件および初期条件を決定し、解析終了時刻、離散化手法、時間刻みなどの実行条件を指定しなければならない。これら一連の、解析前の準備のことをプリプロセッシング、プリプロセッシングを行うためのソフトウェアをプリプロセッサと呼んでいる。本チュートリアルでは、メッシュ作成には OpenFOAM（本チュートリアルにおける使用ソフトウェアの詳細は後述）付属の blockMesh ユーティリティ、および gmshFoam を使用する。境界条件、初期条件、実行条件の設定においては、OpenFOAM の場合、付属のプリプロセッサである FoamX を使用方法と、テキストエディタによる設定ファイルの直接編集およびキーボードからのコマンド入力の組合せで操作する方法の、2 とおりが用意されている。前者の方が習得内容が少なく入門向けであるが、上級ユーザの多くは高度な設定を迅速に行える後者を好む。本チュートリアルでは両方を実施する。

1.7.2 解析実施

解析の準備が整ったら、解析を実行する。この解析を実行するソフトウェアのことを、ソルバと呼ぶ。

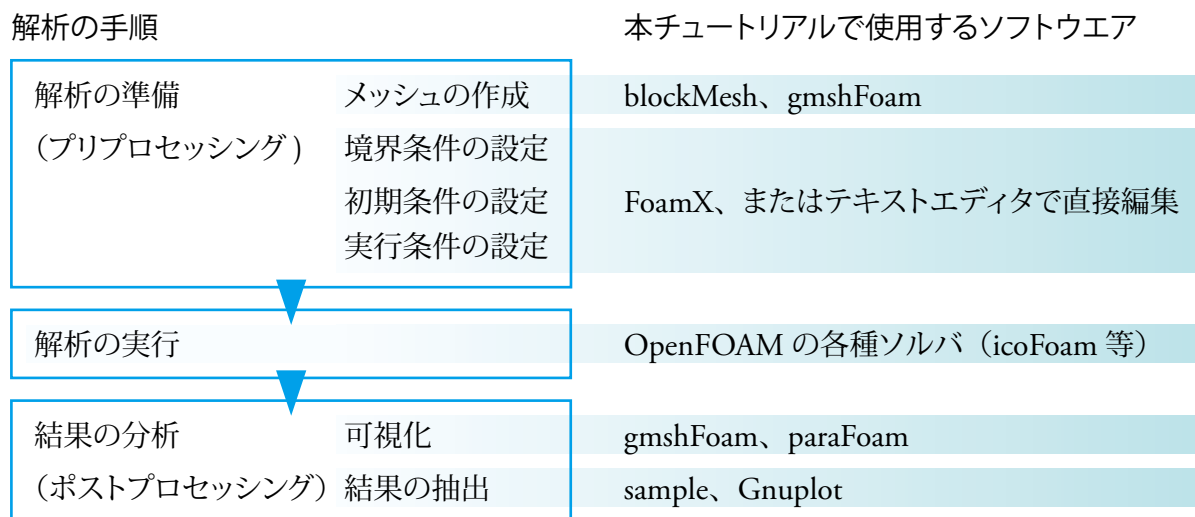


図 1.9 解析実施の手順と使用ソフトウェア

ソルバは、各種の物理現象の支配方程式を前述の離散化手法によってプログラムコード化したソフトウェアであり、数値シミュレーションにおいて最も重要なソフトウェアといえる。OpenFOAM の場合、解析すべき問題の種類に応じてソルバが分かれており、非圧縮性流体解析では icoFoam、弾性体構造解析では solidDisplacementFoam 等と名付けられている。

1.7.3 ポストプロセッシング

解析解析が無事終了したら、結果のデータを分析する。数値シミュレーションの特徴の一つは、解析対象を時空間的に細分割して高解像度のデータが得られることであるが、それゆえに解析結果のデータは大量となる。従って解析結果データそのままの数値で評価することは難しく、多くの場合、コンピュータグラフィックスの技法によって解析対象の形状にデータを重ね合わせ、人間が目で見えて理解しやすいように表現する。この作業のことを可視化と呼んでいる。むろん、解析結果データの一部を取り出して、図 1.2 のような 2 次元グラフにプロットすることも可能である。これらの手法によって解析結果を評価することをポストプロセッシング、ポストプロセッシングを行うためのソフトウェアをポストプロセッサと呼ぶ。本チュートリアルでは、ポストプロセッサとして gmshFoam および paraFoam を使用する。

1.8 本チュートリアルで使用するソフトウェア

本チュートリアルで使用するソフトウェアは、以下各節で述べるものを元としている。これらのソフトウェアは全て、人間が読み書きできる形式であるプログラムコード、すなわちソースコードが無償で公開されているオープンソースソフトウェアである。

1.8.1 Gmsh

ベルギーの University of Liège (リージェ大学) および Catholic University of Louvain (ルーヴァン・カトリック大学) のスタッフによって開発されている、オープンソースソフトウェアとしては珍しいプリ・

ポストプロセッシング両用ソフトウェアである。CAD データおよび独自の形状・メッシュ記述言語によるメッシュ作成機能に重点を置いて開発されている。一方でポストプロセッシング機能も、この種のソフトウェアとしては使用法が非常に簡便でありながら、洗練されたグラフィックス表示を特徴としており、学習用途には最適である。

本チュートリアルで使用する gmshFoam は、Gmsh バージョン 2.0.8 をベースに、OpenFOAM 向けにカスタマイズされたソフトウェアである。

1.8.2 OpenFOAM

ソルバを中心に、プリプロセッシング・ポストプロセッシングのユーティリティをパッケージ化したソフトウェア群である。本チュートリアルでは主にソルバを使用する。従来、英国 Nabla 社の商用シミュレーションソフトウェアであった FOAM (Field Operation and Manipulation の略) が、2004 年 12 月にオープンソース化されたものであり、現在は英国 OpenCFD 社からリリースされている。流体・構造・熱伝導・金融デリバティブ問題、あるいはそれらが組み合わされた連成問題など、非常に多くの問題を解くことが出来るマルチフィジックスコードである。オープンソースであって、なおかつソースコードが非常によく整理されていることから、問題に合わせたソルバを容易に開発可能であることが特徴である。本チュートリアルでも、チュートリアルの目的に合わせて作成されたソルバを使用する。

無償でありながら商用コードに劣らない多機能さから、世界の大学、研究機関はもとより、航空機メーカー・自動車メーカー等、産業界における製品開発の現場でも使用されている。

本チュートリアルでは、OpenFOAM バージョン 1.4 を使用する。

1.8.3 ParaView

米国 Kitware 社からリリースされているポストプロセッサである。開発には Kitware 社のほか、Sandia National Laboratories (米国サンディア国立研究所)、Los Alamos National Laboratory (米ロスアラモス国立研究所)、U. S. Army Research Laboratory (米陸軍研究所) 等の、主に米国の国立研究機関が参加している。最初のリリースは 2002 年 10 月であり、ポストプロセッサとして比較的后発の部類に入るが、活発な開発によって急速にオープンソースポストプロセッサとしてスタンダードの地位を確立した。

本チュートリアルで使用する paraFoam は、ParaView バージョン 3.2.1 をベースに、OpenFOAM 向けにカスタマイズされたソフトウェアである。

1.8.4 Gnuplot

汎用のグラフ作成ソフトウェアである。1986 年に米国ダートマス大学からリリースされて以来、長年にわたって改良が続けられてきた。現在は Gnuplot コミュニティによって開発が続けられており、オープンソースのグラフ作成ソフトウェアとして、ほぼスタンダードである。本チュートリアルでは、Gnuplot バージョン 4.2.3 を使用する。

参考文献

- [1.1] 薩摩順吉、「物理のための数学」、岩波書店（東京）、1995 年
- [1.2] 大矢雅則、岡部恒治ほか、「新編 数学 III」、数研出版（東京）、2005 年
- [1.3] 荒川忠一、「数値流体工学」、東京大学出版会（東京）、1994 年
- [1.4] 葛生和人、「CFD 入門テキスト」、流体物理研究所、2001 年
- [1.5] Christophe Geuzaine and Jean-François Remacle: *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*, <http://www.geuz.org/gmsh/>
- [1.6] OpenCFD Ltd.: *OpenFOAM: The Open Source CFD Toolbox*, <http://www.openfoam.org/>
- [1.7] Kitware Inc.: *ParaView - Parallel Visualization Application*, <http://www.paraview.org/>
- [1.8] *gnuplot homepage*, <http://www.gnuplot.info/>

