GDES利用の手引き

はじめに

GDESはバージョン２になって仕様上空間二次元が計算できることになった．これで計算領域に関する制約が少し緩み大いに計算対象が増えると期待したが，実際には差分法の限界により計算できない問題がかなりあることがわかった．本書の目的はGDESで計算できる範囲を明確にし，間違った計算を行なわないための指針を示すことにある．したがってGDESで計算するための数式モデルを考え，コーディングを行なう人には是非とも理解しておいてほしいことを記述している．まちがった使い方をして適切な計算結果が得られないことはGDESの不当な評価につながり，プログラム開発者としてそのまま見過ごすことのできない問題である．

その一方で，プログラム開発者として可能な限り対応できない問題を減らす改良を行なっていくことは当然のことと考えている．したがって今後もユーザーからの情報を謙虚に受け止め，GDESのバージョンアップを地道に進めていくつもりである．

目次

1. 計算原理

GDESで使用している差分法について説明する．

1.1　離散化

通常，物理量は時間的にも空間的にも連続した値として存在するが，これをデジタル計算機で数値的に模擬する場合，離散値で模擬するのが普通である．GDESでは時間的と空間的に離散化した値で変数値を表わす．

(1)時間方向の離散化（非定常計算の場合のみ）

時間は時間刻み Δt を足し合わせていくことで表現する．つまり $t_j = \sum_{m=1}^{j} \Delta t_m$ である．GDESでは Δt を一定にした計算と Δt を自動調整する計算が可能である．

Δt の自動調整は何ステップかに一回の割合で実施される．自動調整を行なうステップではまず Δt を現状の0.707倍にして２ステップ計算してみる．次に Δt を元の1.414倍にして１ステップ計算を行ない，計算結果を比較する．判定値を越える誤差があれば Δt は元の0.707倍とし，そうでなければ Δt は元の1.414倍とする．

Δt の自動調整を行なうと離散化された時刻をユーザーが想定することはできないが，GDESではtime gridで設定した時刻は必ず評価するようになっている．

時間微分は次の差分式で評価する．

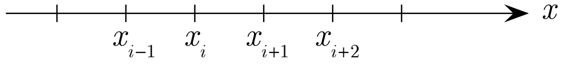
$$\left.\frac{\partial \phi}{\partial t}\right|_{t_j} = \frac{\phi_j - \phi_{j-1}}{\Delta t_j}$$

また，時間的には一次オイラー陰解法にしたがう．つまり形式的には次の式を解く．

$$\frac{\phi_j - \phi_{j-1}}{\Delta t_j} = f\left(\phi_j\right)$$

(2)空間方向の離散化（空間一次元あるいは空間二次元の場合のみ）

一次元の場合，両境界点と有限個の内部点の値で変数値を離散化する．内部点の座標は境界内を等分割するのがもっとも簡単であるが，不等分割した離散化も可能である．



上の格子図で x_i における空間微分は次の三通りの差分式が使い分けられる．

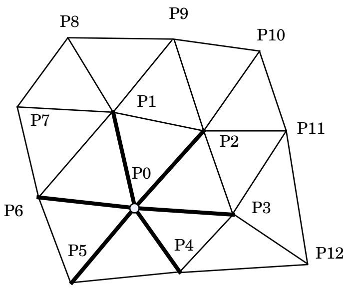
$$\left.\frac{\partial \phi}{\partial x}\right|_{x_i} = C_{i-1}\phi_{i-1} + C_i\phi_i + C_{i+1}\phi_{i+1} \quad \text{（中心差分）}$$

$$\left.\frac{\partial\phi}{\partial x}\right|_{x_i} = C_{i-1}'\phi_{i-1} + C_i'\phi_i \quad （偏った差分１）$$

$$\left.\frac{\partial\phi}{\partial x}\right|_{x_i} = C_i''\phi_i + C_{i+1}''\phi_{i+1} \quad （偏った差分２）$$

ここで C，C'，C'' は離散化された格子座標 x によって決まる定数である．中心差分が二次精度であるのに対し，偏った差分は一次精度であるが，次章で述べるように偏った差分を使うほうがよい場合がある．

二次元の場合，境界線上の点と内部点（いずれも有限個）の値で変数値を離散化する．GDESでは三角メッシュ分割という手順を踏んですべての座標を確定させる．生成された三角形自体に意味はないが，辺の情報は座標と座標の接続情報として利用される．座標と座標の接続は微分項を評価する際に意味を持つ．たとえば次の図のような三角メッシュ分割を例にとると，P0における微分項はP0と三角形の辺でつながった座標（P1, P2，P3，P4，P5，P6）の中の五個の変数値による差分式で評価する[1]．



接続座標が六個以上ある場合はできる限り偏らない五個を自動的に選択する．たとえばP1，P2，P3，P4，P5の五個が選択されたとすると差分式は次のようになる．

$$\left.\frac{\partial\phi}{\partial x}\right|_{P0} = C^x_{P0}\phi_{P0} + C^x_{P1}\phi_{P1} + C^x_{P2}\phi_{P2} + C^x_{P3}\phi_{P3} + C^x_{P4}\phi_{P4} + C^x_{P5}\phi_{P5}$$

$$\left.\frac{\partial\phi}{\partial y}\right|_{P0} = C^y_{P0}\phi_{P0} + C^y_{P1}\phi_{P1} + C^y_{P2}\phi_{P2} + C^y_{P3}\phi_{P3} + C^y_{P4}\phi_{P4} + C^y_{P5}\phi_{P5}$$

接続座標が五個未満の場合，あるいは五個以上の接続座標があるにもかかわらず二次精度の差分式が得られない場合[2]は，できる限り偏らない二個を自動的に選択しそれらの座標の変数値を使った一次精度の差分式で評価する．なおここでは説明を省略するが，境界辺上の点の微分項は内部点の差分式とは異なる近似式を使って評価する．

---

1. 当然P0も使うので実際には六点の変数値による差分式である．

2. 五個以上の接続座標があることが二次精度の差分式が得られるための必要十分条件ではない．たとえば評価位置を含む六点のうち三点が直線上にあり，残りの三点もこれに平行な直線上にある場合，二次精度の差分式は得られない．

【参考】

評価位置とそのまわり５点の変数値を使って微分項を表わすための差分式は次の連立方程式を解いて求める．

$$
\begin{cases}
\phi_{P1} = \phi_{P0} + (x_1 - x_0)\dfrac{\partial\phi_{P0}}{\partial x} + (y_1 - y_0)\dfrac{\partial\phi_{P0}}{\partial y} + \dfrac{(x_1 - x_0)^2}{2}\dfrac{\partial^2\phi_{P0}}{\partial x^2} + \dfrac{(y_1 - y_0)^2}{2}\dfrac{\partial^2\phi_{P0}}{\partial y^2} + (x_1 - x_0)(y_1 - y_0)\dfrac{\partial^2\phi_{P0}}{\partial x\,\partial y} \\
\phi_{P2} = \phi_{P0} + (x_2 - x_0)\dfrac{\partial\phi_{P0}}{\partial x} + (y_2 - y_0)\dfrac{\partial\phi_{P0}}{\partial y} + \dfrac{(x_2 - x_0)^2}{2}\dfrac{\partial^2\phi_{P0}}{\partial x^2} + \dfrac{(y_2 - y_0)^2}{2}\dfrac{\partial^2\phi_{P0}}{\partial y^2} + (x_2 - x_0)(y_2 - y_0)\dfrac{\partial^2\phi_{P0}}{\partial x\,\partial y} \\
\phi_{P3} = \phi_{P0} + (x_3 - x_0)\dfrac{\partial\phi_{P0}}{\partial x} + (y_3 - y_0)\dfrac{\partial\phi_{P0}}{\partial y} + \dfrac{(x_3 - x_0)^2}{2}\dfrac{\partial^2\phi_{P0}}{\partial x^2} + \dfrac{(y_3 - y_0)^2}{2}\dfrac{\partial^2\phi_{P0}}{\partial y^2} + (x_3 - x_0)(y_3 - y_0)\dfrac{\partial^2\phi_{P0}}{\partial x\,\partial y} \\
\phi_{P4} = \phi_{P0} + (x_4 - x_0)\dfrac{\partial\phi_{P0}}{\partial x} + (y_4 - y_0)\dfrac{\partial\phi_{P0}}{\partial y} + \dfrac{(x_4 - x_0)^2}{2}\dfrac{\partial^2\phi_{P0}}{\partial x^2} + \dfrac{(y_4 - y_0)^2}{2}\dfrac{\partial^2\phi_{P0}}{\partial y^2} + (x_4 - x_0)(y_4 - y_0)\dfrac{\partial^2\phi_{P0}}{\partial x\,\partial y} \\
\phi_{P5} = \phi_{P0} + (x_5 - x_0)\dfrac{\partial\phi_{P0}}{\partial x} + (y_5 - y_0)\dfrac{\partial\phi_{P0}}{\partial y} + \dfrac{(x_5 - x_0)^2}{2}\dfrac{\partial^2\phi_{P0}}{\partial x^2} + \dfrac{(y_5 - y_0)^2}{2}\dfrac{\partial^2\phi_{P0}}{\partial y^2} + (x_5 - x_0)(y_5 - y_0)\dfrac{\partial^2\phi_{P0}}{\partial x\,\partial y}
\end{cases}
$$

上記の連立方程式を解くと $\dfrac{\partial\phi_{P0}}{\partial x}$，$\dfrac{\partial\phi_{P0}}{\partial y}$ 以外に $\dfrac{\partial^2\phi_{P0}}{\partial x^2}$，$\dfrac{\partial^2\phi_{P0}}{\partial y^2}$，$\dfrac{\partial^2\phi_{P0}}{\partial xy}$ の差分式も同時に求まる．この連立方程式はMathematicaなどの数式処理ソフトを使って解くと簡単である．

1.2　連立方程式の組み立てと解法

与えられた基礎式をすべての格子点（離散化された空間点）上で離散化変数を使って記述すると（未知変数の数×格子点数）を元数とする連立方程式が得られる．この連立方程式は元の基礎式が非線形であればやはり非線形方程式となるため非線形連立方程式の解法が必要である．GDESではニュートン法によって線形化を行ない，線形化された連立方程式を直接法で解いている．ニュートン法で必要となるヤコビ行列は微小変位による数値微分を使用している．

2. 適用範囲

GDESは前章までで説明した差分法により計算を行なう．したがって差分法の特性をよく理解した上で実際の問題に適用する必要がある．

2.1　差分法の特徴

差分法の基本的な規則は次のとおりである．

①空間／時間に分布する値を格子上の値の集合で表現する．

②微分項は格子上の値によるテイラー級数展開の有限項を使って近似する．

これらの単純な規則だけによって偏微分方程式を解くので次のような特性を持っている．

ⓐ汎用性が高い．いかなる微分方程式も（解が得られるかどうかは別として）コード化できる．

ⓑ計算の安定性に関しては何の保証もない．

ⓒ微視的な近似計算なので全体を積分した値には累積誤差が伴う．

【中心差分（二次精度）と偏った差分（一次精度）】

求めようとする解がテイラー級数の主要項でよく近似できる場合は中心差分が偏った差分より精度がよいことは言うまでもない．しかし，必ずしもテイラー級数展開が解をうまく近似しない場合がある．典型的な例が移流方程式である．一般的に言うと，**二方向性座標上の計算では中心差分**がよく，**一方向性座標上の計算では上流側に偏った差分**がよい[1]．

2.2　空間一次元

空間一次元の場合，差分法に起因する欠点はほとんどカバーする手段がある．したがってGDESで計算できない問題はあまりない．

結論を先に言うと，コントロールボリュームの考え方を導入することで安定性に関する問題と積分量の誤差の問題が解消する．たとえば次のような保存量 ϕ に関する支配方程式を考えてみる．

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial x}\left(\phi u\right) = S \qquad \text{(i)}$$

単純にGDESコードで表わすと，

```
%equ:phi[t]+phi*u[x]+phi[x]*u-S
```

となる．これでも計算結果が得られる場合が多いが，二つの問題がある．一つ目は計算結果に振動成分が発生することである．これが安定性の問題であり，条件によっては発散して解が得られなくなる．この問題だけ

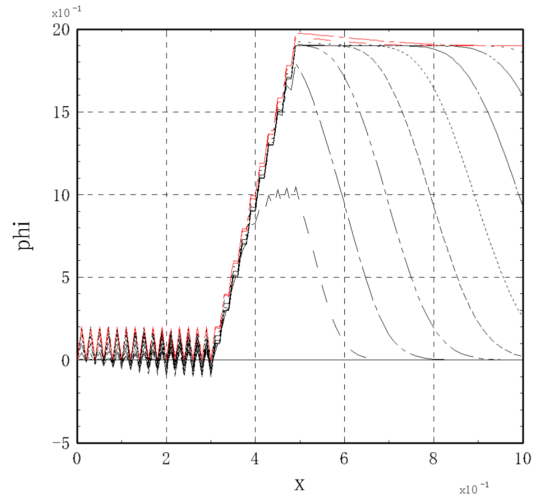
---

1. 二方向性座標とはある座標の状態がその両側の状態の影響を受けるような座標のことで，たとえば一次元熱伝導現象がこれに該当する．他方，一方向性座標とはある座標の状態がその片側の状態の影響しか受けないような座標のことで，たとえば時間が典型的な例である他に移流現象もこれに該当する．詳細は「コンピュータによる熱移動と流れの数値計算」（スハス　V，パタンカー著，森北出版）P. 20参照．

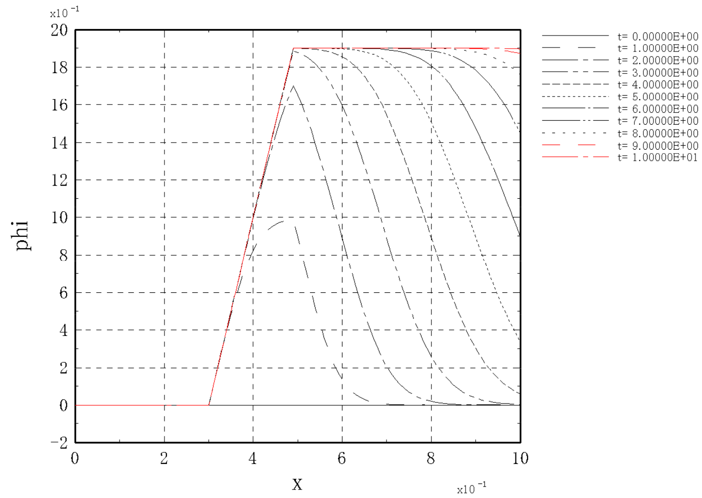
ならば上流差分を使用することで解決することができる．たとえば $u > 0$ の場合，前述のコードは次のように書き換えられる．

```
%equ:phi[t]+phi*u[x,0]+phi[x,0]*u-S
```

同じ微分方程式の中心差分による結果と上流差分による結果の一例を次に示す．



中心差分

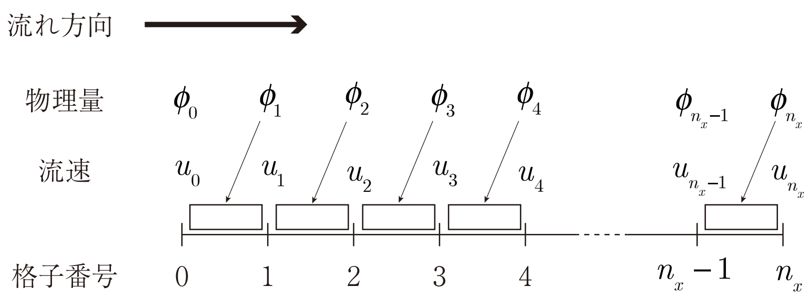


上流差分

二つ目の問題は積分量の誤差である．元の支配方程式は保存則を表わしているわけだから，計算によって得られた数値解も完全に保存していることが望ましい．ところが差分法では格子点だけに着目した式を取り扱うため，当初の値が完全には保存していないことがある．この問題を解決するためにGDES上で次節に示す方法でコントロールボリューム法[1]を模擬する．

・コントロールボリューム法

GDESでは格子上に定義された物理量が実は格子間のコントロールボリュームに定義された離散値であるとみなす（下図）．



1. 有限体積法とも言う．計算領域を微小有限要素（コントロールボリューム）に分割し，要素内の積分量と要素からの出入り量に着目して保存則を計算する方法．要素内の保存と計算領域全体の保存が常に満足される．

コントロールボリューム内の物理量の分布は一様とみなし，コントロールボリューム間に速度を定義する．これは流体計算でよく使われるスタガード格子である．こうすると格子i-1と格子iの間のコントロールボリュームにおける物理量 ϕ の収支は次の式で表わすことができる．

$$\left(x_i - x_{i-1}\right)\frac{\partial \phi_i}{\partial t} = \phi_{i-1}u_{i-1} - \phi_i u_i + S_i\left(x_i - x_{i-1}\right) \tag{ii}$$

ここで $\boldsymbol{\Phi} = \phi u$ とおくと，次の式が得られる．

$$\frac{\partial \phi_i}{\partial t} + \frac{\boldsymbol{\Phi}_i - \boldsymbol{\Phi}_{i-1}}{x_i - x_{i-1}} = S_i \tag{iii}$$

この式はGDESでは次のコードで模擬できる．

```
%equ:phi[t]+phiu[x,0]-S
%equ:phiu-phi*u
```

つまり，一次元空間においてはコントロールボリューム法による離散式と差分法（上流差分）による離散式はたまたま一致することから，GDESを使ってコントロールボリューム法と完全に同じ計算が可能であり，保存量の全体誤差の問題も解消できるのである．ただしこの方法には数値拡散による精度の低下という欠点がある．つまり上流からきた情報がコントロールボリューム内全体に一瞬で伝わるという現実離れしたモデルなのである．

コントロールボリューム法を使う場合の注意すべき点をまとめると次のようになる．

①格子上に割り当てた変数の実際の位置関係に十分注意すること[1]

②物理量以外に物理流束を未知変数として定義する必要があること

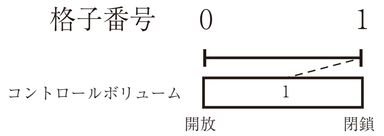
③コントロールボリューム内の物理量が完全に一様であるという非現実的な仮定を前提にしていること

1. このことから流れ方向は常に同方向でなければならないといえる．部分的にでも逆流する現象は計算できない．

◇**増圧による計算領域内への流入現象**◇

一次元空間の片側が閉鎖，その逆端が開放されている場合，領域内に満たされた気体は圧力の変化に応じて開放端で流入あるいは流出が起こる．たとえば増圧すれば流入があり，減圧すれば流出がある．実際にGDESによるコントロールボリューム法で減圧による流出現象は計算できる．ところがこの計算方法では**増圧による流入現象は計算できない**．ここにその理由を示す．

要点を明確にするために一個のコントロールボリュームで一次元空間が表わされているとする．右端が閉鎖端で左端が開放端なら増圧によって左端から流入があり，領域内の流速は右向きになるはずである．したがって上流差分の考え方に従えばコントロールボリューム内の物理量は格子番号１に定義されなければならない．



このような計算格子上におけるガス密度とガス流速に関する連立方程式は次のようになる．

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \dfrac{\partial}{\partial x}\left(\rho u\right) = 0 \\ PM = \rho RT \end{cases}$$

ここで，ρ はガス密度，u はガス流速，P は全圧，M はガス分子量，R はガス定数，T はガス温度である．このうち P，M，R，T は他の式で決まったり与えられる値であるから，ρ は輸送方程式とは関係なく決まることが容易にわかる．つまり残る未知変数 u が輸送方程式で計算されることになる．

ところが実際に格子ごとに境界条件を考慮して差分式に置き換えようとすると困ったことが起こる．コントロールボリューム１の収支計算を格子番号１上の輸送方程式で評価するので，閉鎖端で流速を $u_1 = 0$ に拘束する格子が存在しないことになる．逆に開放端では特別な境界条件を課さないが，このときGDESは上流側格子の存在しない上流差分の代わりに下流側の差分式を使用する．格子番号０における下流側の差分は格子番号１における上流差分と完全に等しい．つまり格子番号０における輸送方程式と格子番号１における輸送方程式が完全に同一式になってしまうのである．

結果的には右端格子における境界条件 $u_1 = 0$ を左端格子上の輸送方程式の代わりに課すことができれば問題は起こらないのであるが，GDESの仕組み上，左端格子上で右端格子上の未知変数を引用することができないため計算不可能である[1]．

1. 特殊な方法になるが，右端の流速を保持する未知変数を追加することで計算可能になる．

2.3　空間二次元

空間二次元になると差分法では解けない問題が増えてくる．次のような問題は計算できないと考えておいたほうが無難である．

(1)空間一方向性座標問題（たとえば流れ場を含むもの）←前述した差分法の特性ⓑに起因する

(2)全体の保存を完ぺきに計算しなければならない問題（コントロールボリュームの考え方を導入する必要がある問題）←前述した差分法の特性ⓒに起因する

したがってGDESで計算できるものは次のような問題に限定される．

①拡散方程式／熱伝導方程式のように拡散項／熱伝導項が支配的な問題．ただし全体の保存量が近似的に成り立っていればよいもの．

②既知の流れ場内のスカラー保存則．ただし全体の保存量が近似的に成り立っていればよいもの．

簡単に言うと空間的には，**二方向性座標では計算できる**が，**一方向性座標では計算できない**ということになる[1]．

【計算できる例】

熱伝導計算，拡散計算，クリープ流れ計算，ポテンシャル流れ計算，既知の流れ場内のスカラー保存計算

【計算できない例】

NS方程式

【参考】NS方程式の特殊な場合であるクリープ流れの計算方法を示す．流れ場を表わす一般的な基礎式は次のとおり連続の式と運動方程式である[2]．

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \boldsymbol{u}\right) = 0 \quad \text{（連続の式）}$$

$$\left.\begin{aligned} \frac{\partial}{\partial t}\left(\rho u\right) + \nabla \cdot \left(\rho u \boldsymbol{u} - \mu \nabla u\right) &= -\frac{\partial p}{\partial x} \\ \frac{\partial}{\partial t}\left(\rho v\right) + \nabla \cdot \left(\rho v \boldsymbol{u} - \mu \nabla v\right) &= -\frac{\partial p}{\partial y} \end{aligned}\right. \quad \text{（運動方程式）}$$

ここで非圧縮の定常流れに限定すると次のように簡略化できる．

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{（連続の式）}$$

$$\left.\begin{aligned} \nabla \cdot \left(\rho u \boldsymbol{u} - \mu \nabla u\right) &= -\frac{\partial p}{\partial x} \\ \nabla \cdot \left(\rho v \boldsymbol{u} - \mu \nabla v\right) &= -\frac{\partial p}{\partial y} \end{aligned}\right. \quad \text{（運動方程式）}$$

1. 一方向性座標上の保存計算を行なう場合はGFlowを使う方がよい．

2. 運動方程式に剪断応力項を加えたものがNS方程式である．

さらに粘性係数が大きく，流れが緩やかで移流項が無視できる[1]とすると運動方程式は次のようになる．

$$
\begin{aligned}
-\nabla\cdot\left(\mu\nabla u\right) &= -\frac{\partial p}{\partial x}\\
-\nabla\cdot\left(\mu\nabla v\right) &= -\frac{\partial p}{\partial y}
\end{aligned}
\quad \text{（運動方程式）}
$$

粘性係数一定として書き直すと結局解くべき式は次のようになる．

$$
\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \quad \text{（連続の式）}
$$

$$
\begin{aligned}
\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) &= \frac{\partial p}{\partial x}\\
\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right) &= \frac{\partial p}{\partial y}
\end{aligned}
\quad \text{（運動方程式）}
$$

これをこのままの形でGDESで解こうとしてもうまくいかない．次に連続の式を圧力のポアソン方程式に変形する．まず，二つの運動方程式をそれぞれ x ， y で微分して次の式を得る．

$$
\begin{aligned}
\mu\frac{\partial}{\partial x}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) &= \frac{\partial^2 p}{\partial x^2}\\
\mu\frac{\partial}{\partial y}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right) &= \frac{\partial^2 p}{\partial y^2}
\end{aligned}
$$

この二つの式を足すと圧力のポアソン方程式が得られる．

$$
\mu\left(\frac{\partial^3 u}{\partial x^3}+\frac{\partial^3 u}{\partial x\,\partial y^2}+\frac{\partial^3 v}{\partial x^2\,\partial y}+\frac{\partial^3 v}{\partial y^3}\right)=\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}
$$

この式は次のように変形できる．

$$
\mu\left\{\frac{\partial^2}{\partial x^2}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)+\frac{\partial^2}{\partial y^2}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\right\}=\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}
$$

これに連続の式を代入すると，

$$
\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}=0
$$

が得られる．実際の計算ではこれと同等の次の式を使う．

1. 言い換えるとレイノルズ数がほぼ0であるということ．

$$\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}=D\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)$$

ここで D はなるべく大きい値とする．以上をまとめるとクリープ流れを解くための基礎式は次のとおりである．

$$\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}=D\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)$$

$$\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)=\frac{\partial p}{\partial x}$$
$$\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)=\frac{\partial p}{\partial y}$$

これを見てわかるようにこれらの基礎式は熱伝導方程式などと同類の「楕円型方程式」である．二次元空間の微分方程式のうち計算できるものは「楕円型方程式」に限定されると言ってよい．参考のためこの基礎式に対応したスクリプトの例を次に示す[1].

```
# (nx,ny):法線方向（外向き）
nx=attribute(3)
ny=attribute(4)
%if:attribute(1)~1 # non-slip
  equ_p=nx*p[x]+ny*p[y]
  equ_u=u
  equ_v=v
%else if:attribute(1)~2 # inlet
  equ_p=nx*p[x]+ny*p[y]
  equ_u=-u*nx-v*ny-uin
  equ_v=u*ny-v*nx
%else if:attribute(1)~3 # outlet
  equ_p=nx*p[x]+ny*p[y]+p
  equ_u=nx*u[x]+ny*u[y]
  equ_v=nx*v[x]+ny*v[y]
%else if:attribute(1)~4 # free-slip
  equ_p=nx*p[x]+ny*p[y]
  equ_u=u*nx+v*ny
  equ_v=nx*(ny*u[x]-nx*v[x])+ny*(ny*u[y]-nx*v[y])
%else
  equ_p=p[x,2]+p[y,2]-(u[x]+v[y])*10000
  equ_u=p[x]-mu*(u[x,2]+u[y,2])
  equ_v=p[y]-mu*(v[x,2]+v[y,2])
%end if
%equ:equ_p
%equ:equ_u
```

1. この方法はかなり特殊な計算方法である．そもそも誤差評価が正確に行なえないという重大な欠点がある．やはり流れ場計算はGFlowなどのコントロールボリューム法を使うことが望ましい．

%equ:equ_v

3. 特殊な計算

3.1　空間座標を時間発展的に解く方法

たとえば空間三次元であっても，二次元形状の押し出しで形作られる三次元空間であり，なおかつ物質移動が完全な押し出し流れであれば，GDESの時間軸を押し出し方向空間軸と見なした計算が可能である．この方法は，押し出し流れ方向を z ，押し出し流れ速度を w としたときに移流項が $\frac{\partial}{\partial z}\left(\phi w\right)$ だけで表わせる場合に限定される．つまり $u = v = 0$ でなければならない．また，押し出し方向の拡散項／熱伝導項が無視できない場合は計算できない．なぜなら拡散や熱伝導は境界条件を与えて二方向座標で計算しなければならない問題だからである．GDES空間軸方向だけの拡散項／熱伝導項はあってもよい．

GDESで空間軸を時間発展的に解く場合の必要条件は次のとおりである．

(1)計算形状は，時間軸に置き換える空間軸方向に同じ形状（押し出し形状）であること．

(2)時間軸に置き換える空間軸方向以外の移流項が無視できること．

(3)時間軸に置き換える空間軸方向の拡散項／熱伝導項が無視できること．

この方法によって三次元空間の定常計算が行なえる[1].

3.2　複数の未知変数順列で空間座標を表現する方法

GDESの空間格子を使用しないで，すべての空間上の未知変数を別々に定義し，空間微分項を基礎式としてコーディングする方法．理論的にはいかなる計算も可能であるが，現実的には次のようなケースが限界と思われる．

(1)未知変数の順列で表わす空間の格子数は５～20ぐらいまで．[2]

(2)未知変数の順列で表わす空間の格子は等間隔．[3]

なお，空間微分項の離散化の方法としては狭義の差分法またはコントロールボリューム法のいずれを使うことも可能であるが，一般的にはコントロールボリューム法の離散式のほうが複雑になると考えておくべきである．特に極座標系ではその違いが顕著になる．

---

1. この方法を使った二次元空間や一次元空間の定常計算ももちろん可能である．

2. 格子数に比例してコーディング量が増加する．同じ式のコピーと小修正ではあるが，20分割ぐらいが労力的に限界と考えられる．

3. 不等間隔の場合，差分式が複雑になり，コーディング量が増加する．不等間隔格子にするより等間隔格子として格子数を増やすほうがよい場合がある．

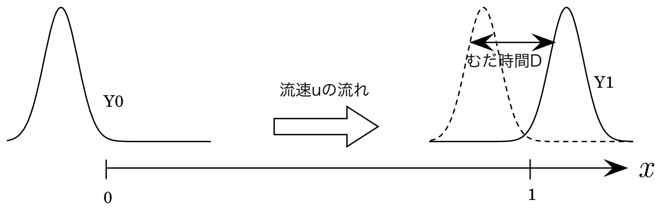
3.3　むだ時間（time delay）を模擬する方法

むだ時間は制御系でよく出てくるモデルで，入力値が一定時間遅れてそのまま出力されるものである．非常に単純な概念である割りに数式モデルが難しいもののひとつである．GDESを使って制御系の微分方程式を解こうとしたときには，おそらくこのむだ時間が大きな問題となる．

幸い制御系では空間分割が不要である場合がほとんどである．そこで仮想的な空間を設けて情報量が空間を伝達していく時間をむだ時間とする考え方が可能である．つまり，情報量が一次元の一定速度の流れに乗って終点まで流れていくと考え，始点での変数値に対してむだ時間を与えた変数値が終点の値と見なすのである．ただし，通常の流れの計算には数値粘性が混ざってしまうので，厳密なむだ時間を再現するためには工夫が必要である．具体的な手順を示すため次のように想定する．

・むだ時間を適用する前の未知変数をY0，Y0にむだ時間を適用した未知変数をY1とする．

・むだ時間はDとする．



(1) 始点における未知変数の値をそのまま終点に流す方法

理論的には変数 ϕ が速度 u で流れていく現象は，次の輸送方程式を解けばよいはずである．

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = 0$$

しかしこのままGDESで計算すると前に述べたように数値粘性の影響によって完全な輸送計算にならない．その理由はこの微分方程式から導かれる差分式を見るとわかる．上流差分式を使ってこれを書き直すと次のようになる．

$$\phi_i^{t+\Delta t} = \phi_i^t - \frac{u\Delta t}{\Delta x}\left(\phi_i^{t+\Delta t} - \phi_{i-1}^{t+\Delta t}\right)$$

これではいかにしても $\phi_i^{t+\Delta t} = \phi_{i-1}^t$ なる式は得られない．完全な輸送計算にするためには差分式が次のようになっている必要がある．

$$\phi_i^{t+\Delta t} = \phi_i^t - \frac{u\Delta t}{\Delta x}\left(\phi_i^t - \phi_{i-1}^t\right)$$

この場合，$\dfrac{u\Delta t}{\Delta x}$ が１のとき，$\phi_i^{t+\Delta t} = \phi_{i-1}^t$ となる．u はむだ時間Dからおのずと決まる値（$u = \dfrac{1}{D}$）であるが，Δt と Δx には自由度があるから $\dfrac{u\Delta t}{\Delta x} = 1$ という条件で計算を行なうことは可能なはずである．したがって，この差分式を実現させるために１時間ステップ前の ϕ が引用できればよいということである．それではここで１時間ステップ前の ϕ を保持する φ という未知変数を導入することにする．

$$\varphi = \phi - \Delta t \frac{\partial \phi}{\partial t}$$

これは差分式では次のようになる．

$$\varphi_i^{t+\Delta t} = \phi_i^{t+\Delta t} - \Delta t \frac{\phi_i^{t+\Delta t} - \phi_i^t}{\Delta t}$$

ゆえに $\varphi_i^{t+\Delta t} = \phi_i^t$，つまり φ は１時間ステップ前の ϕ に等しい．

以上より，完全な輸送方程式を再現する差分式とそれと同等な微分方程式は次のようになる．

$$\phi_i^{t+\Delta t} = \phi_i^t - u\Delta t \frac{\varphi_i^{t+\Delta t} - \varphi_{i-1}^{t+\Delta t}}{\Delta x}$$

$$\frac{\partial \phi}{\partial t} + u \frac{\partial \varphi}{\partial x} = 0$$

これでY0の値をそのまま終点に流す方法はわかった．次に終点のY0（始点のY0に対してちょうどむだ時間Dだけ遅れた値）を始点で参照する方法を説明する．

(2) 終点のY0を始点でY1として参照する方法

これは非常に簡単で，終点側の境界条件Y0=Y1として次の微分方程式を解くだけである．

$$\frac{\partial Y_1}{\partial x} = 0$$

(3) まとめ

Y0に対してむだ時間Dを適用したY1を模擬するための手順をまとめると次のようになる．

①Y0の支配方程式を始点に記述する．

②全空間座標でY0の１ステップ前の値を保持する未知変数Y0pを定義する．

③始点を除く空間座標にY0の輸送方程式を記述する．このとき移流項はY0pを使った微分とする．

④終点に%equ:Y1-Y0を記述する．

⑤終点を除く全空間に%equ:Y1[x, 1]を記述する.

スクリプト例を示す.

```
%if:attribute(1)~1
  equ_Y0=Y0-sin(t)
  equ_Y1=Y1[x,1]
%else if:attribute(1)~2
  equ_Y0=Y0[t]+u*Y0p[x,0]
  equ_Y1=Y1-Y0
%else
  equ_Y0=Y0[t]+u*Y0p[x,0]
  equ_Y1=Y1[x,1]
%end if
%equ:equ_Y0
%equ:equ_Y1
%equ:Y0p-Y0+delt*Y0[t]
```

3.4　コントロールボリューム法を模擬する際の逆流対処法

・逆流で発散する理由

コントロールボリューム法の模擬方法の節で示した離散式(ii)は $u \geq 0$ を前提としている．この前提がなければ(ii)式は次のように記述しなければならない．

$$\left(x_i - x_{i-1}\right)\frac{\partial \phi_i}{\partial t} = \max\left(0, u_{i-1}\right)\phi_{i-1} + \min\left(0, u_{i-1}\right)\phi_i - \max\left(0, u_i\right)\phi_i - \min\left(0, u_i\right)\phi_{i+1} \qquad \text{(ii-a)}$$

つまり $u < 0$ になると右辺に第二項あるいは第四項が表われてくるがこれをGDESで正確に記述することが難しい．そのまま第一項や第三項で代用して計算しようとすると数値安定性の問題で発散してしまうのである．

・対処方法

(ii-a)式の右辺第二項あるいは第三項を評価するときに問題となるのは着目格子以外の格子上の変数値を使わなければならないことである．つまり $u < 0$ の場合は $\boldsymbol{\Phi}_i = \phi_{i+1} u_i$ というように格子iと格子i+1の変数値の演算が必要である．

これをGDESで計算しようとすると次のようにして隣接格子上の変数値を取得して計算を行なうようにする[1]．

```
%if:u<0
  phi_0=phi+phi[x,1]*system(7)
%else
  phi_0=phi
%end if
%equ:phiu-phi_0*u
```

【解説】

system(7)　　$x_{i+1} - x_i$

phi[x,1]　　$\dfrac{\phi_{i+1} - \phi_i}{x_{i+1} - x_i}$

$\therefore$ phi$_i$+phi[x,1]*system(7)=phi$_{i+1}$

1. このように隣接格子上の値を得るためには該当変数が「未知変数」である必要がある．「サブ変数」ではこの方法で隣接格子上の値を得ることができないため他の手段が必要である．もっとも手っ取り早いのはすべて「未知変数」に変更する方法であるが，未知変数が増加することによる行列計算速度の低下は避けられない．もうひとつの方法はサブ変数定義式を隣接格子上の値を取得する必要のあるすべての場所に記述する方法である．計算速度の低下はほとんどないが，コーディング量が増えてめんどうである．

4. その他のヒント

4.1　条件分岐ブロック内で%equを使った間違い

GDESはスクリプト内の方程式を「%equ」という識別子で識別する．このような単純な規則で方程式を抽出し連立方程式を組み立てるので，何らかの条件によって方程式を使い分ける場合は注意が必要である．たとえば次のような式を解きたいとする．

$$\begin{cases} H = 1.24T^2 + 135T, \text{ if } T < 100 \\ H = 900T, \text{ if } T \geq 100 \end{cases}$$

ここで H は未知変数，T はインプットなどの既知の値とする．

これを次のようなスクリプトで表現するのは間違いである．

```
%if:T<100
  %equ:H-1.24*T*T-135*T
%else
  %equ:H-900*T
%end if
```

T の値によって評価される式が切り替わるところまでは想定通りであるが，方程式の数が二つと認識されてしまうため想定通りの結果が得られない．このようなコーディングを行なってしまった場合，他に間違いがなければ未知変数の数と方程式の数が一致していないというエラーメッセージにたどり着く．正しいコードは次のとおりである．

```
%if:T<100
  v=H-1.24*T*T-135*T
%else
  v=H-900*T
%end if
%equ:v
```

未知変数 H に対する支配方程式は最終行の方程式一つである．GDESはローカル変数vが0になるように H を計算するのである．

4.2　不適切な式

数学的には同等であっても数値計算上不適切となる式がある．たとえば次の式の評価を考える．

$$\alpha = \frac{1}{e^T}$$

ここで T が絶対温度のような正の値とわかっている場合，この式を次のスクリプトで表現することは好ましくない．

```
alpha=1/exp(T)
```

組み込み関数expは引数が少し大きな値になるとすぐにオーバーフローエラーを起こす．このような式は次のようなスクリプトで表現するべきである．

```
alpha=exp(-T)
```

あるいは次のような式を考える．

$$\alpha = \frac{1}{\frac{1}{u} - 1 + u}$$

これをこのまま次のようなスクリプトで表現すると，u が 0 になることが予想される問題の場合に収束が悪くなったり，最悪の場合収束しなかったりする．

```
alpha=1/(1/u-1+u)
```

この場合も同等な次のスクリプトで記述すると問題なくなる．

```
alpha=u/(1-u+u^2)
```

これら二つのケースは，最初の例が $\frac{1}{\infty}$，後の例が $\frac{1}{0}$ という計算不可能な項が部分的に現われるものであるが，いずれも式の変形で回避できる．このほかの $\frac{0}{0}$ や $\frac{\infty}{\infty}$ といった不定形についても同様である．

4.3 クランク-ニコルソン法／陽解法など

時間微分を含む方程式は次の差分式に離散化される．

$$\frac{\phi_j - \phi_{j-1}}{\Delta t_j} = f\left(\alpha\phi_j + \left(1-\alpha\right)\phi_{j-1}\right)$$

通常 $\alpha = 1$ でユーザーがこの値を変更することはできないが，GDSファイルを直接編集することで α を変更することができる．GDSファイルの中に次の一行がある[1]ので所望の値に書き換える．

```
%ALPHA value
```

たとえばvalue=0.5とするとクランク-ニコルソン法が模擬できるし，value=0とすると陽解法が模擬できる．この方法で α を変更する場合は次のことに注意しなければならない．

1) 時間微分項を含まない方程式がある場合は $\alpha = 0$ としてはならない（計算できない）．

2) 計算結果として得られる未知変数値はあくまでも t_j における値であるが，サブ変数値は $\alpha t_j + \left(1-\alpha\right)t_{j-1}$ における値になる．

---

1. まれに%ALPHAの行がない場合もあるがその場合は一行追加する．

3) %ALPHAの行を書き換えるとその後のファイルの保存時には常に書き換えられた値が継承される．もとの値１に戻したい場合は再度GDSファイルを直接編集する必要がある．

GDESがデフォルトで $\alpha = 1$ を採用している理由は，時間という一方向性座標に対して完全陰解法が望ましいこともあるが，2) による違和感を避けるためであることも大きい．

4.4　想定外の時間刻み

たとえば時間刻みを0.1にして計算終了時刻を1.0で非定常計算を行なうとする（ユーザーによる時間格子の追加はないものとする）．このとき計算を行なう者は，0.0，0.1，0.2，0.3，…，0.8，0.9，1.0の合計11ステップの計算が行なわれるだろうと予想するのが普通であろう．ところが実際には次の12ステップの計算が行なわれる．

0.0，0.1，0.2，0.3，0.4，0.5，0.6，0.7，0.8，0.9，0.95，1.0

・時間格子に自動的に加えられる計算終了時刻

計算原理の章で説明したようにGDESは時間格子（time　grid）を使って時間軸方向の計算を進める．実はコントロールデータ画面のtime　gridで指定した時刻だけでなく計算終了時刻も時間格子に加えられる．つまり計算終了時刻はユーザーが指定した時刻であるから計算刻みの積分結果に関わらず必ず計算すべき時刻であるという考え方である．したがってこの例では1.0という時刻だけが時間格子に指定されていることになる．

・丸め誤差

ところでユーザーが指定した計算刻み0.1は内部表現では循環小数となるため丸め誤差を含んでおり $\frac{1}{10}$ よりわずかに小さい．したがって計算刻み0.1で10ステップ進んだとき時刻は1.0にわずかに達せず1.0－ε となっている．GDESは計算格子に指定されている計算終了時刻1.0では必ず計算しようとするので，時刻1.0－ε と計算終了時刻1.0の間で調整を行ない時刻0.95（前ステップと時間格子時刻の中間）と計算終了時刻1.0で計算を行なうのである[1]．

ちなみに0.1を10回足した1.0と直接指定した1.0のIEEE倍精度内部表現は次のようになる．

（0.1の足し算）　　4607182418800017407

（1.0直接指定）　　4607182418800017408

以上のことからもともと計算刻みに丸め誤差が含まれていないか丸め誤差があっても大きい値に丸められている場合にはこの問題は起こらない．いずれにしても時間格子による調整は必ず精度を向上する方向に行なわれるので問題になることはないと考えている．

1. 時刻1.0－ε と時刻1.0の両方で計算を行なうことがあまり意味のないことはおわかりいただけるであろう．