

COMSOL 技術特別セミナー 2016年7月8日(金) 10:00-11:30

COMSOL Multiphysics における数値解析について

橋口真宜

第1技術部部長

計測エンジニアリングシステム株式会社

東京都千代田区内神田 1-9-5 井門内神田ビル 5F

<http://www.kesco.co.jp>,

<https://www.comsol.jp>

注：本書は講演会当日の内容に加筆修正したものである。

1. PDE (偏微分方程式) について

質量保存則、運動量保存則、エネルギー保存則 (非相対論的な意味) は、ある物理量 $u(x,t)$ (スカラとする) の時間微分と、その物理量に伴う流束(Flux) Γ (ベクトル量とする) の空間微分の形に記述できる。

これらの式は一般に、

$$\frac{\partial u}{\partial t} + \nabla \cdot \Gamma = f$$

と記述できる。

この式の形は一般性があり、いろいろなフィジックスに適用できる。

2. 汎用的な数値解法としての有限要素法と行列解法

上の式を数値的に解くことができれば、広範囲なフィジックスの数値解を得ることができる。

2. 1 重み付き残差法

有限要素法は、重み付き残差法を基礎にしている。つまり、上の式を空間の各点で正確に満たすような数値解を得ることは困難であるので、数値解を上のに代入したときの方程式の残差 (もし数値解が正解であれば残差は 0) を考えている空間で平均的に 0 にすることを考える。そのために、重み関数 ϕ を導入し、それを上の式の両辺に掛け、それを考えている空間 Ω で積分を行う (積分=平均のことと思えばよい) と、2次元では $\Gamma=(u,v)$ として次の関係

$$\nabla \cdot \phi \Gamma = \nabla \cdot (\phi u, \phi v) = (u, v) \cdot \nabla \phi + \phi \nabla \cdot (u, v) = \Gamma \cdot \nabla \phi + \phi \nabla \cdot \Gamma$$

を使うと、

$$\int_{\Omega} [\phi \frac{\partial u}{\partial t} + \phi \nabla \cdot \Gamma - \phi f] d\Omega = \int_{\Omega} [\phi \frac{\partial u}{\partial t} + \nabla \cdot (\phi \Gamma) - \Gamma \cdot \nabla \phi - \phi f] d\Omega = 0$$

Gauss の発散定理を使うことで、 \mathbf{n} を境界上に立てた外向き単位法線ベクトルとすると、

$$\int_{\Omega} \nabla \cdot (\phi \Gamma) d\Omega = \int_{\partial \Omega} \phi \Gamma \cdot \mathbf{n} dS$$

を得る。このように一部、境界での積分に移行する。一般に境界での流束は流入流束で考えたほうが良いので、この両辺に負号を付けたものを組み込めるように整理すると、

$$\int_{\partial \Omega} -\phi \Gamma \cdot \mathbf{n} dS + \int_{\Omega} \left[\Gamma \cdot \nabla \phi + \phi f - \frac{\partial u}{\partial t} \right] d\Omega = 0$$

を得る。この式では元の式にあった Γ の微分 ∇ が外れているために、 Γ に関する微分の条件が緩和されている。ここで導出した式は弱形式(weak form)と呼んでいる。もとの微分方程式は強形式(strong form)と呼ばれている。

有限要素法は、弱形式に現れる境界積分、体積積分ともに、数値積分を行う。解析対象としている領域を有限個の要素で覆いつくし、各要素に適切に未知数 u_i を配置する。未知数同士を接続する形状関数は空間座標の関数として与えることで未知数 u の微分を形状関数の空間微分に置換する。この部分を空間の離散化と呼ぶことにする。

未知数 u_i の数を M 個とする。すると M 個の独立した式が必要であるが、重み関数 ϕ は任意に設定してよいので、 M 個の重み関数 ϕ を考えることで、必要な M 個の独立した式を構築できる。それによって、 M 個の連立方程式を M 個の未知数 u_i について構築できたことになり、あとはそれを数値計算によって解けば良いということになる。境界条件としてノイマン条件(例:ゼロ流束では、 $-\mathbf{n} \cdot \Gamma = 0$)を使う場合には、境界積分は既知となり、上述の連立方程式の右辺に現れる列ベクトルとなる。他方、ディリクレ条件(例: $u=0$)の場合には、そのままでは取扱いにくいので、境界値 u_b を満たすような反力 \mathbf{R} を導入し、反力を未知数として扱うことにする。ラグランジュ乗数の導入に相当する。

したがって、 $\mathbf{Ax}=\mathbf{b}$ を如何にメモリ・計算時間ともに効率よく解くかということが議論されることになる。

2. 2 行列の解法について

COMSOL Multiphysics はデフォルトでフィジックスに応じたソルバ(行列解法)を提案してくれるが、何かのときのために、一応のことは知っておいたほうがよい。

2. 2. 1 直接解法

(1) 行列 A の LU 分解とガウスの消去法

L は下三角行列(対角成分の右上にある要素が全て 0)、 U は上三角行列(対角成分の左下にある要素が全て 0) であるとする。このとき、係数行列 A を

$$A=LU$$

のように L と U の積に分解できたとする。これを A の LU 分解と呼ぶ。

すると、

$$Ax=LUx=b$$

となり、

$$Ly=b$$

を y について解き、

次に、

$$Ux=y$$

を x について解くことで解 x を求めることができる。

L と U は共に三角行列であるので、解くのは簡単である。

$Ly=b$ はガウスの消去法の前進代入に相当し、 $Ux=y$ は同じく後退代入に相当する。

ガウスの消去法と LU 分解は連立一次方程式を有限回の操作で解く方法であり、直接法と呼ばれている。

例)

$$\mathbf{A} = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

これを LU 分解すると、以下を得る。

$$\mathbf{L} = \begin{pmatrix} 1.00000 & 0.00000 & 0.00000 & 0.00000 \\ -0.50000 & 1.00000 & 0.00000 & 0.00000 \\ 0.00000 & -0.66667 & 1.00000 & 0.00000 \\ 0.00000 & 0.00000 & -0.75000 & 1.00000 \end{pmatrix}$$

$$\mathbf{U} = \begin{pmatrix} 2.00000 & -1.00000 & 0.00000 & 0.00000 \\ 0.00000 & 1.50000 & -1.00000 & 0.00000 \\ 0.00000 & 0.00000 & 1.33333 & -1.00000 \\ 0.00000 & 0.00000 & 0.00000 & 1.25000 \end{pmatrix}$$

b を以下のようにする。

$$\mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$$

まず、 $Ly=b$ を解くと、

$$\mathbf{y} = \begin{pmatrix} 1.0000 \\ 2.5000 \\ 4.6667 \\ 7.5000 \end{pmatrix}$$

続いて、 $Ux=y$ を解くと、

$$\mathbf{x} = \begin{pmatrix} 4.0000 \\ 7.0000 \\ 8.0000 \\ 6.0000 \end{pmatrix}$$

を得る。

A が正定値対称行列の場合については、コレスキー分解(Cholesky 分解)を利用した方法がある。これは

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T = \mathbf{U}^T\mathbf{U}$$

の形に分解した後に解を求める方法である。

例)

$$A = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \end{bmatrix}$$

$$L = \begin{bmatrix} 1.73205 & 0.00000 & 0.00000 \\ 1.15470 & 1.29099 & 0.00000 \\ 0.57735 & 1.03280 & 1.26491 \end{bmatrix}$$

L は $1.73205 = \sqrt{3}$ など平方根を含む。

このように、この方法では対角要素の計算に平方根の計算が必要である。その計算を不要にした計算法として、一般の対称行列についての修正コレスキー法がある。対角行列を D としたとき、

$$A = LDL^T$$

と分解する方法である。

(2) A の逆行列

$$A^{-1} = (LU)^{-1} = U^{-1}L^{-1}$$

の関係を利用して求める。

2. 2. 2 反復解法

定常反復解法： Jacobi 法、Gauss-Seidel 法、SOR 法

優対角（対角要素の絶対値が非対角要素の値に比べて大きい）で疎な（0 要素がたくさんあるような）係数行列をもつ連立一次方程式 $Ax=b$ の係数行列 A を次のように分解する。

$$A = D + E + F$$

ここで、 D は対角要素のみの行列、 E は下三角行列（対角要素なし）、 F は上三角行列（対角要素なし）。

$$Ax = (D + E + F)x = b$$

このとき、次の反復公式を作ることができる。

$$x^{k+1} = -D^{-1}(E + F)x^k + D^{-1}b$$

これはヤコビ法(Jacobi 法)である。

例)

$$A = \begin{bmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{bmatrix}$$

```

b = 1
    1
    1
    1

D = 4  0  0  0
    0  4  0  0
    0  0  4  0
    0  0  0  4

E =  0  0  0  0
    -1 0  0  0
     0 -1 0  0
     0  0 -1 0

F = 0  -1  0  0
     0  0 -1  0
     0  0  0 -1
     0  0  0  0

```

初期ベクトル $x_0=0$ としてヤコビ法を実行してみる。

```

x1 = 0.25000
     0.25000
     0.25000
     0.25000

```

x_0 に x_1 を代入して、新しい x_1 を計算すると、

```

x1 = 0.31250
     0.37500
     0.37500
     0.31250

```

十数回目で

```

x1 = 0.36363
     0.45454
     0.45454
     0.36363

```

となり、正確な値に収束する。

ヤコビ法で、解 x^{k+1} が求まったらずぐにそれを使う方法がガウス・ザイデル法 (Gauss-Seidel) である。

$$x^{k+1} = D^{-1}(b - Ex^{k+1} - Fx^k)$$

つまり、

$$x^{k+1} = -(D + E)^{-1}Fx^k + (D + E)^{-1}b$$

を得る。

例)

この場合だと、7回目 (Jacobi 法の半分の回数) で正確値に収束する。

SOR 法 (Successive over-relaxation) は加速緩和法ともよばれる方法であり、 x^{k+1} をそのまま使わずに、1 より大きい ω (加速パラメタ) を使って、 $\omega(x^{k+1}-x^k)$ を補正量とする方法である。

$$\begin{aligned}\xi^{k+1} &= D^{-1}(b - Ex^{k+1} - Fx^k) \\ x^{k+1} &= x^k + \omega(\xi^{k+1} - x^k)\end{aligned}$$

実際の計算では、 ω として 1 より小さい数値を使うこともある。その場合、不足緩和(Under relaxation)と呼ぶ。

これらの反復法はいずれも

$$x^{k+1} = Mx^k + Nb$$

の形で表される。 M は反復回数 k に依存しないので定常反復解法と呼ばれる。

この反復法が収束するための必要十分条件は、「 M の固有値の絶対値が全て 1 より小さい」ということである。

先ほどの A についてヤコビ法の M の固有値を求めてみると、

$$\begin{aligned}A &= \begin{matrix} 4 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{matrix} \\ e &= \begin{matrix} -0.40451 \\ -0.15451 \\ 0.15451 \\ 0.40451 \end{matrix}\end{aligned}$$

となっていていずれの固有値の絶対値も 1 より小さい。 A が優対角行列であれば、反復行列 M の固有値の絶対値はすべて 1 より小さい。つまり、必ず収束する。

ガウス・ザイデル法ではこの場合、

$$\begin{aligned}e_1 &= \begin{matrix} 0.00000 \\ 0.02387 \\ 0.16363 \\ 0.00000 \end{matrix}\end{aligned}$$

となっている。固有値の縮退があればその分、収束が速くなる。

SOR 法については正定値対称行列については $0 < \omega < 2$ であれば収束することが示されてい

る。ガウス・ザイデル法で収束しない場合には SOR 法が使える。並列計算ではガウス・ザイデル法は使えないのでヤコブ法や SOR 法が使える。

非定常反復解法：共役勾配法 (Conjugate Gradient)、双共役勾配法 (BiCG)、二乗共役勾配法 (CGS)、安定化双共役勾配法 (Bi-CGSTAB)、Spectrally Accelerated BiCGStab(SA)、一般化最小残差法 (GMRES) ほか

これらは、反復ごとに変化する情報を積極的に計算に取り込む非定常的な解法である。

拘束や最適化条件が加わる。クリロフ部分空間 (Krylov 部分空間) への写像を基底として使用するため、Krylov 部分空間法ともよばれる。

Krylov 部分空間法では、一般に近似解の列 $x^0, x^1, x^2, \dots, x^{n-1}$ が

$$x^k = x^0 + z^k$$

$$z^k \in K_k(A; r^0)$$

を満たすように生成する。

ある固定した r^0 に $n \times n$ 行列 A を乗じて生成されるベクトル $r^0, Ar^0, A^2r^0, \dots, A^{k-1}r^0$ が互いに 1 次独立であり、これらのベクトルによって張られる k 次元のクリロフ部分空間 $K_k(A; r^0)$ に z^k は属する。このとき、残差は

$$r^k = b - Ax^k = r^0 - Az^k$$

であり、この残差 r^k も $K_{k+1}(A; r^0)$ に含まれる。

行列 A が対称の場合には r^0 から出発するランチョス過程によって残差 $\{r^k\}$ が $K_n(A; r^0)$ の直交系をなすようにし、 x^k が決まるようにした反復解法が共役勾配法 (CG) である。 A が対称でない場合にはクリロフ部分空間における直交系をつくるためにはアーノルド過程を利用しなければならない。この方法はランチョス過程よりも計算量が増大するために、工夫を施すことになる。

一般化残差法 (GMRES) は、Gram-Schmidt 直交化法に基づいているが、CG 法のように簡単な 3 項漸化式で書き表せないために、反復回数が増えるにつれてベクトル列と上ヘッセンベルグ行列を保存しておく必要がある。これは計算量およびメモリの点から実用的ではない。そこで記憶容量を減らしたリスタート版 GMRES が考案された。

双対共役勾配法 (Bi-CG) は非エルミート行列を係数行列にもつ連立一次方程式に対する解法である。

2. 2. 3 反復解法と前処理

前処理：不完全 LU (ILU)、ポイントヤコビ法 (Point Jacobi)、Smoothed Aggregation 代数型マルチグリッド (SA-AMG) ほか

共役勾配法は効率を上げるために様々な前処理が提案され、現在では前処理付共役勾配法は大規模な連立一次方程式を解く最も実用的な方法の一つになっている。

方程式 $Ax=b$ の係数行列 A の固有値に縮退があると、共役勾配法は真の解に到達するま

での反復回数が減少する。固有値が縮退しているものとして単位行列 I がある。そこで、 $Ax=b$ を共役勾配法で解く際に、元の連立方程式に適切な前処理を施して単位行列を係数にもつ方程式に変換した後に共役勾配法を適用することを考える。

係数行列 A を不完全に修正コレスキー分解する前処理を不完全コレスキー分解共役勾配法 (ICCG 法 : Incomplete Choleski decomposition Conjugate Gradient method) と呼んでいる。

2. 2. 4 マルチグリッド法

ヤコビ法とガウス・ザイデル法は反復過程で滑らかな誤差を生成する。その誤差の高周波数成分は複数回の反復後に除去できるが、低周波数成分はなかなか減衰しない。

そこで、反復計算の過程で細かい格子と粗い格子の間を行き来しながら誤差を効率よく減衰させるようにしたものがマルチグリッド法である。格子は複数の階層を使う。行き来の仕方によって V サイクルや W サイクルといった呼び名がある。

いろいろな手法があり、多階層の離散化を行なう点に特徴がある。

MIT の資料を見てみよう。

<http://ocw.mit.edu/courses/mathematics/18-086-mathematical-methods-for-engineers-ii-spring-2006/readings/am63.pdf>

ここでは 1 の長さの区間を格子幅 $h=1/8$ で分割する。境界条件は両端 ($x=0, x=1$) で 0 とする。求めたい解は内部の $u_1, u_2, u_3, u_4, u_5, u_6, u_7$ である。マルチグリッド法で使う粗い格子としては 2 倍の格子幅をもつ $2h=1/4$ とし、その格子上の解を v_1, v_2, v_3 とする。

Restriction R: 細かい格子から粗い格子への解のコピー

$$v_0 = u_0 = 0$$

$$v_1 = u_2$$

$$v_2 = u_4$$

$$v_3 = u_6$$

$$v_4 = u_8 = 0$$

Interpolation (or Prolongation) I: 粗い格子から細かな格子への線形補間

$$u_1 = 1/2(v_0 + v_1) = v_1/2$$

$$u_3 = 1/2(v_1 + v_2)$$

$$u_5 = 1/2(v_2 + v_3)$$

$$u_7 = 1/2(v_3 + v_4) = v_3/2$$

2. 3 COMSOL における反復解法

2. 3. 1 解法と特徴

CG (Conjugate gradients)

これは $Ax=b$ で A が正定値で対称な行列であるような線形系に使う反復解法である。行列が正定値でない場合でも正定値に近い場合にもこのソルバは機能する。このソルバは使用メモリが少なく、GMRES ソルバより速いことが多い。しかし、ある限られたモデルについてのみ適用される。速い収束を得るには適切な前処理を使うことが重要である。

BiCGStab, (biconjugate gradient stabilized)

$Ax=b$ の形の一般的な線形系のソルバである。反復一回あたりのメモリと計算時間は一定である。つまり、反復回数が増えても計算時間とメモリは一定である。BiCGStab は GMRES が 2 回の反復に使うメモリとほぼ同じ量を使う。BiCGStab は GMRES よりも少ないメモリを必要とする。

BiCGStab の収束性は GMRES よりもより不規則になりがちである。途中の残差は初期残差よりも数桁大きくなり得る。アルゴリズムが残差の精度が悪いことを検出したら、初期見積もりとして現時点の解を使って反復を再スタートさせる。GMRES、CG とはちがって、BiCGStab は反復ごとに 2 回の行列ベクトルの掛け算を実行する。これは反復ごとに 2 回の前処理を必要とする。また、左前処理付 BiCGStab を使うときは、反復ごとに前処理が追加される。従って、左前処理付 BiCGStab は反復ごとに 3 回の前処理を行なうことになる。

GMRES (Generalized minimum residual)

これは線形系のソルバであり、リスタート GMRES 法を使う。 $Ax=b$ の形をした一般的な線形系に関する反復解法である。速い収束を得るには、適切な前処理を使うことが重要である。

FGMRES (Flexible generalized minimum residual)

これは GMRES の変形であり、ロバストな方法で広い範囲の前処理を扱うことができる。FGMRES に関してはどんな前処理を使っても良い。一方で、GMRES の 2 倍のメモリを食う。FGMRES は右前処理を使うので右前処理付 GMRES と同じ収束判定基準を有する。FGMRES が不完全 LU 前処理のような一定前処理を用いる場合には、FGMRES は右前処理付き GMRES と同じものになる。

反復解法の収束判定

$$\rho |M^{-1}(b - Ax)| < \text{tol} \cdot |M^{-1}b|$$

これは、「相対残差×因子 ρ はトレランス tol よりも小さければ反復過程を終了」することを意味する。 M が単位行列であるソルバでは行列 A が悪条件である場合には不正確な

解であっても早い段階で計算を終了してしまう。M が単位行列ではないソルバでは、M がプアな前処理であれば反復計算を早期に終わってしまうことが多い。そのような場合には因子 ρ を行列 $M^{-1}A$ の条件数の程度の大きさをもつ数値まで増やすようにする。 ρ がその条件数よりも大きい場合には、この収束判定は相対誤差が tol よりも小さい、つまり、

$$|x - A^{-1}b| < \text{tol} \cdot |A^{-1}b|$$

であることを示すことになる。

この判定が満たされたら、ソフトウェアは計算を終了して解を戻す。判定を満たさない場合には解法は停止し、エラーメッセージを出力する。

M はソルバによって以下のように定義されている。

M=LU

MUMPS, PARDISO, SPOOLES

M=前処理行列

GMRES, CG, BiCGStab を左前処理とともに使う場合

2. 3. 2 反復線形系ソルバの前処理の選択

反復線形系ソルバを使う場合には前処理を選ばなくてはならない。

前処理は反復回数と収束に影響を与える。前処理は時間とメモリを食う。

前処理としては、反復ノードを右クリックして必要なものを選ぶ。

PRECONDITIONER	USAGE
GENERAL FRAMEWORK	
Multigrid — Geometric multigrid	For elliptic or parabolic systems.
Multigrid — Algebraic multigrid and smoothed aggregation AMG	For scalar problems or loosely coupled multiphysics problems of the elliptic or parabolic type.
Domain Decomposition	For large problems in a distributed-memory system or as an alternative to a direct solver.
Krylov Preconditioner	For Helmholtz problems where the mesh does not fulfill the Nyquist criteria. It can be used on the coarse multigrid level or as a smoother.
FULL OR APPROXIMATE FACTORIZATION OR APPROXIMATE INVERSE	
Incomplete LU	For nonsymmetric systems (the default preconditioner).

PRECONDITIONER	USAGE
Sparse Approximate Inverse (SAI)	For BEM methods and as a general preconditioner or smoother. It has a costly setup phase but typically shows a good parallel scalability.
Direct Preconditioner	For small fields (an ODE, for example), where a direct solver is efficient.
POINTWISE GENERAL PURPOSE	
SOR	For elliptic problems without zeros on the diagonal. Typically better than Jacobi and still rather inexpensive.
Jacobi (diagonal scaling)	For large positive definite models.
BLOCK GAUSS-SEIDEL GENERAL PURPOSE	
SCGS	For fluid-flow problems with linear elements.
SOR Line	For the same problem class as for SOR but adopted to stretched/anisotropic meshes (for example, boundary layer meshes). More expensive than SOR.
Vanka	For large indefinite problems with zeros on the diagonal of the system matrix.
VECTOR ELEMENT METHODS	
Auxiliary Space Maxwell (AMS)	For curl-curl problems stemming from stationary or time-dependent Maxwell's equations.
SOR Gauge	For ungauged vector element formulations of Magnetostatics.
SOR Vector	For large electromagnetics models using vector elements.

3. 空間の離散化と連立方程式の関係

未知数の空間配置、形状関数の選択、数値積分の手法が、得られる連立方程式の係数行列の内容を決める。

3. 1 ガラーキン法

これは重み関数 ϕ の形状関数を、未知数 u_i の形状関数と同じものにする。COMSOL Multiphysics は、従属変数を u としてモデルビルダを構成した場合、重み関数は $\text{test}(u)$ と記述できる。これはガラーキン法の特徴をとらえた非常にうまい表現である。重み関数 ϕ の x 微分は、 $\text{test}(u_x)$ と書く。

3. 1. 1 楕円型、放物型の PDE

数値安定性の問題が無い場合には、PDE インターフェースをそのまま適用すれば良い。

3. 1. 2 数値安定性の問題があるもの

流体問題であり、流線拡散、ガラーキン最小二乗法 (GLS)、横風安定 が組み込まれている。モデルビルダの表示で 離散化にチェック を入れると、離散化セクションに現在の状態が表示される。

3. 1. 3 ラグランジュ要素、エッジ要素、Serendipity 他

PDE インターフェースの離散化セクションで、

Lagrange, Hermite, Discontinuous Lagrange, Nodal discontinuous Lagrange, Discontinuous scalar density, Divergence, Bubble, Gauss point data, Nodal serendipity がリストとして確認できる。

3. 1. 4 ガウス点積分法

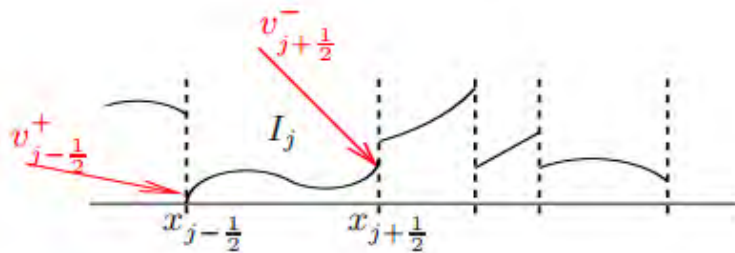
ガウス点積分法は、積分点の位置を最適化し、ベストフィットになるような重みをあらかじめ決めておくことで少ない積分点数で精度のよい積分を行うことができる。

FEM でのベクトルや行列を構成するために使っている。

3. 2 節点不連続ガラーキン法(dG)

連続ガラーキン法(CG: Continuous Galerkin)とは異なって、不連続ガラーキン法(DG: Discontinuous Galerkin)は区間連続のみを満たす試行関数を利用する。有限要素法と有限体積法の混合したような方法である。要素の中には通常の有限要素と同じく多項式近似を利用するが、要素間をつなぐ対流項は有限体積法のように風上化した数値流束を使うといったことが行われる。DG は 1970 年初めに開発されたが、1990 年初めから CFD で使われるようになった。

DG の説明図を以下に示す。このように区間と区間の境界は不連続を許す。空間の離散化は DG で行い、時間方向には陽的 Runge-Kutta 法を使うといったことも行われている。



Discrete space:

$$V_h = V_h^k = \{v : v \in P^k(I_j), \forall j\}$$

Some notations:

$$I_j = [x_{j-1/2}, x_{j+1/2}], \quad \Delta x_j = |I_j|, \quad \Delta x \text{ or } h = \max_j |I_j|$$

$$v_{j+1/2}^- = \lim_{\epsilon \rightarrow 0^-} v(x_{j+1/2} + \epsilon), \quad v_{j-1/2}^+ = \lim_{\epsilon \rightarrow 0^+} v(x_{j-1/2} + \epsilon)$$

$$[v]_{j-1/2} = v_{j-1/2}^+ - v_{j-1/2}^-$$

不連続ガラーキン法 (DG) は、有限要素法のもつ幾何形状の柔軟な取扱いと、有限体積法のもつ強い非線形波動方程式の取り扱いならびに高度な並列性を、共に利用しようとする技術である。DG は電磁波問題や非線形弾性問題などへも応用されている。

3. 2. 1 波動問題

波動形式 PDE インターフェースでは、波動方程式を時間一階、空間に不連続 Galerkin 法を使っている。計算速度とメモリ使用量を高度に最適化している。

3. 2. 2 完全時間陽的な取り扱い

ルンゲ・クッタ法とアダムス・バッシュフォース法は時間陽的ソルバである。節点不連続 Galerkin 法は陽的時間ソルバを用いる場合に自然でもっとも効率的な解法である。

ほかの状況でメリットがある場合としては、粒子追跡あるいは波動問題のいずれかを使う場合がある。

3. 2. 3 行列フリー、積分フリー

波動形式 PDE は、陽的時間積分を志向している。この方法は行列も積分も使わない。要素での局所行列のみを構成するので、時間積分幅を自動的に設定できる。行列を構成しなければメモリ使用量を低減できる。

3. 2. 4 流体の条件設定との関係

COMSOL では、壁一すべり条件、周期的流れ条件、流れ連続性、という境界条件で使われている。COMSOL では、対称内部ペナルティ Galerkin 法 (SIPG Symmetric Interior Penalty Galerkin) である。この方法は、点ごとよりは積分の意味で境界条件を満たす。

3. 3 境界積分の離散化 (BEM)

3. 3. 1 グリーン関数が既知の場合

PDE の係数は定数になる。

3. 3. 2 八分木

Octree は木構造の一種で、各ノードに最大 8 個のノードがある。

階層で行列を表現できる。高速化に貢献する。

3. 3. 3 Adaptive cross approximation (ACA)

この方法は、メモリと CPU 時間を短縮できる。表面積分を使うモーメント法 (MoM) に使われているような方法である。

3. 3. 4 行列フリー反復解法

BiCGStab/SAI (SAI: Sparse approximation inverse)

SAI は前処理であり、系の行列の逆行列を近似する。SAI のメリットは、行列とベクトルの掛け算の箇所に前処理を適用できる点である。これは並列処理でも効率がよい。構築する時間はかかるが、並列処理とスケーラビリティにも効果がある。ILU あるいは SOR ほどには効率は良くはないが、SAI は非等方なメッシュや演算のような特定の問題に適していることがよくわかっている。BEM 法においては有用であることもわかっている。

4. 非線形問題、定常

COMSOL_ReferenceManual.pdf(Ver.5.2a)の p.773 に非線形ソルバの設定に関する記述がある。

非線形問題では、ソルバのロバスト性とパフォーマンスを改善するためにソルバのデフォルト設定を変更することもあり得る。

(1) ロバスト性の改善

非線形ソルバが収束しない場合は次の方法を試す。

1) 自動高度非線形性 (ニュートン) を使って非常に小さい減衰ファクタで計算を開始させそのファクタを徐々に増加させる、あるいはある小さい一定値の減衰ファクタを使う。減衰ファクタは 0 と 1 の間の数値である。0.1 という数値は非常に小さい減衰ファクタを意味し、ロバストではあるが収束は遅い。減衰ファクタを小さくした場合には非線形反復の回数を増やす必要がある。一定の小さい減衰ファクタを使っても非線形ソルバが不安定な場合には自動高度非線形性を使って自動的に減衰ファクタを非常に低い数値から増加させるとよい。

2) よい初期見積もり (予想される解、境界条件と整合のとれた解) を使う。ソルバの補助スイープ機能や時間依存でのステップ関数を使ったランピング (ある数値から徐々に目的値まで変化させる方法) を使う。簡単な問題 (例えば温度依存のない解)、一方向連成といったものから計算を開始させるのも良い。初期条件を境界条件に合わせるよりは、境界条件を更新する方法が簡単で有ったりもする。

3) よい初期見積もりが用意できない場合、各分離ステップで小さい減衰ファクタを使った分離解法が収束を助けることがある。

4) ヤコビアンを各反復ごとに毎回更新することで、非線形ソルバは方程式系から最適な情報を入手できる。これは、非線形性が温度自身である場合、例えば材料が温度に依存する場合といったときや、自然対流のように温度と同じグループで計算される他の変数 m に依存する場合には必要である。

(2) 収束速度の改善

収束速度が遅い場合には次の方法を試す。

1) 線形問題では一定減衰ファクタとして 1 を使う。ログを見れば線形問題であるかどうか分かる。

2) よい初期値を用意する。

3) 収束について言えば、完全連成ソルバは分離ソルバよりも良い収束速度を持つ。

4) 最小ヤコビアン更新を使うとヤコビアンの計算に使う時間を節約できる。この方法は、線形や適当な非線形性をもつモデルに適している。

4. 1 完全連成解法 (拘束なし)

デフォルトではヤコビアンの更新は最小に設定されている。ニュートン非線形法はデフ

オルトでは、定常解析では自動減衰ファクタ計算、時間依存解析では一定ダンピングファクタとなっている。

長所：

良い初期見積もりがあれば速い非線形収束を示す。

短所：

ヤコビアン構成あるいは計算にコストがかかる

4. 2 分離解法（拘束なし）

デフォルトでは、伝熱との連成では伝熱の従属変数を別の分離グループにする。

長所：

行列構成の時間が短い。線形系をうまく扱える。

短所：

非線形収束速度が遅い

4. 3 定常問題、拘束あり

完全連成解法

4. 4 定常問題、拘束の処理法

ヌル空間行列を探す。

二つの別個の問題に帰着させる。

4. 5 定常解法

4. 5. 1 完全連成

ニュートン法の減衰付きを利用する。初期減衰ファクタは小さくしている。これは、初期値が最終解に近くない場合にフルのニュートンステップが悪さをし得るからである。非線形アルゴリズムは自動的に減衰ファクタを調整して収束解に到達する。

一定/自動減衰（ニュートン法）

詳細なモデルでは自動減衰ニュートン法は十分なロバスト性を示さないかもしれない。その場合、擬時間ステップアルゴリズムが使えるであろう。

ダブルドッグレグ法（ニュートン法とコーシー方向行列の組み合わせ）

ダブルドッグレグ法は定常問題に利用できる。これはニュートン信頼領域法の一つであり、非線形方程式を解くとき、方向を調整できる。

4. 5. 2 分離解法

グループ化した変数群の間の反復

各グループについて、ニュートン法を適用

安定化と加速 (これらはオプション)

擬時間ステップング (局所の人為的な時間ステップの PID 制御、 CFD で使用)

$$\rho(\mathbf{u} \cdot \nabla)\mathbf{u} = \nabla \cdot [-p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)] + \mathbf{F}$$

$$\rho \frac{\mathbf{u} - \text{nojac}(\mathbf{u})}{\Delta t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} = \nabla \cdot [-p\mathbf{I} + \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)] + \mathbf{F}$$

アンダーソン加速 (探索方向の履歴に基づいて残差を最小化)

- 4. 6 初期値問題に見られる一階時間微分の離散化 (拘束なし)
- 4. 7 BDF 離散化
- 4. 8 初期値問題に見られる一階時間微分の離散化 (拘束あり)
- 4. 9 拘束のタイプ 1 (Eliminate)
- 4. 10 拘束のタイプ 2 (Eliminate-Lagrange)
- 4. 11 陽的ルンゲ・クッタ法 (q 段階法)
- 4. 12 初期値問題に見られる二階時間微分の離散化
- 4. 13 一般化 α
ニューマーク法の説明
- 4. 14 一般化 α
二次精度
- 4. 15 波動問題の一階微分系への書き直しと BDF, RK などの適用
- 4. 16 BDF の適用の続き
- 4. 17 陽的 RK 法の適用

- 5. 時間依存ソルバ
 - 5. 1 BDF ソルバ (SUNDIALS/IDAS)
1 次から 5 次精度の陰的、アダプティブ、感度解析、イベント
イベント(event)とは、そのイベントが生じたら、グローバル状態を再初期化する機能
のことである。
 - 5. 2 一般化 α
2 次精度、陰的、調節可能な減衰、アダプティブ、 粒子追跡
 - 5. 3 ルンゲ・クッタ
1 次から 4 次精度、陽的、半アダプティブ
 - 5. 4 ルンゲ・クッタ 34(Soenderlind), 45(Dormand-Prince, Cash-Karp)

- 3次、5次精度、陽的、アダプティブ、硬直性検出
- 5. 5 アダムス・バッシュフォース
 - 3次精度、陽的、半アダプティブ、局所時間ステップング
- 5. 6 時間離散
 - CFD プロジェクション法
- 5. 7 モーダル解法
 - 固有関数を使った次数低減系の解析
 - 目的は、固有関数を使うことで、モデルの低減を行ない、あるシミュレーションの高速化を図る。
 - いま、取り扱っている系が次式で記述できるとする。

$$E\ddot{u} + D\dot{u} + Ku = L$$

E は質量行列、 D は減衰行列、 K は剛性行列、 L は荷重ベクトル。
 モーダル解析アルゴリズムは、何個かの固有ベクトルが計算されていることを前提としている。このとき、解を少数の固有関数で表現できるとすると、

$$u_m = \Phi_r q$$

と書くことができる。

$$E_m \ddot{q} + D_m \dot{q} + K_m \Phi_r q = \Phi^H L,$$

$$\text{where } E_m = \Phi^H E \Phi, D_m = \Phi^H D \Phi, \text{ and } K_m = \Phi^H K \Phi.$$

詳細は省略するが、これによって、各固有値方向に独立な少数の方程式を解けば良いということになる。

6. 標準スタディの行列の問題

- 6. 1 定常、完全連成
- 6. 2 時間依存、完全連成

(1) BDF (Backward Differential Formulas)

1次精度(後退オイラー法)から5次精度まで用意。

この方法は、長時間の計算に使う。問題も安定とわかっている場合に使う。

この方法は、特に低次精度の方法において、強い減衰効果がある。後退オイラー法はどんな高周波数の誤差も減衰させることができる。鋭い時間変化を期待する場合に後退オイラー法を使えばなまった波形になってしまう。

(2) 一般化 α

二次の BDF と基本的には同じ考え方であるが、内容は異なる。 α というパラメタを含んでおり、高周波数の減衰度合いを制御する。2次精度までの BDF と比べると、一

一般化 α は減衰が小さく、より正確である。ただし、安定性は BDF よりも劣る。COMSOL での一般化 α は、時間一階微分の変数、時間二階微分の変数を自動的に検出し、適切な公式を当てはめる。ほとんどの場合、一般化 α は十分な安定性をもつ正確な方法である。輸送計算といった多くのフィジックスでは COMSOL は一般化 α をデフォルトの時間依存ソルバとして設定している。しかし、ある複雑な問題ではよりロバストな BDF をデフォルトに設定している。

(3) より高精度のスキーム

常微分方程式(ODE)のように、陽解法であるルンゲ・クッタ法が偏微分方程式の時間依存問題にも使える。RK34、Cash-Karp 5、Dormand-Prince 5 が最も効率的である。

6. 3 固有値 (f は固有値探索位置)

6. 4 周波数領域 (f は与えられた周波数)

7. 線形ソルバに関するまとめ

COMSOL_Reference Manual.pdf の p.773 に線形ソルバのチューニングに関する記述がある。収束しない、あるいは収束速度が遅い場合に考える。

(1) PARDISO への切り替え

マルチグリッド(MG)の前処理として GMRES がデフォルトで設定されている。その場合はメモリ(RAM)容量が十分であれば、PARDISO に切り替えてみる。PARDISO はメモリをたくさん使うが、通常、収束が良い。PARDISO が収束しないときは、モデルの設定が悪いのか、他のソルバ設定に問題があるかもしれない。

(2) GMRES/マルチグリッドのメモリ

反復ソルバに使うメモリをさらに最適にするには粗いレベルのマルチグリッド用の格子上のメッシュ要素数を減らすことができる。例えば、メッシュ粗さファクタ、あるいはマルチグリッドレベルを増やす。マルチグリッドレベルを増やすと計算時間はかかる。

(3) GMRES/マルチグリッドの収束性

線形ソルバの収束が悪い場合には、次の方法を考える。

1) GMRES の収束グラフが 50 回の反復数毎にゆっくりした収束を示す場合には、リスタート前の反復数 (デフォルトは 50) を増やす。通常、倍にしてみる。これはメモリを増やすことにはなる。

2) マルチグリッドの反復回数を増やすと、プリ smoother、ポスト smoother での、前処理と GMRES の収束が改善される。

3) 2 つのマルチグリッドレベルの間の差が大きいと収束に影響するので、マルチグリッドのメッシュ粗さファクタを小さくすると、収束性を改善できるかもしれない。

4) マルチグリッドレベルのメッシュを手動で作成することを考える。

7. 1 定常、時間依存問題

単一フィジックス方程式は、しばしば良い性質を持っている。

最良の線形ソルバ(MG)を使うことができる。

分離解法はマルチフィジックスを扱う際に単一フィジックスの線形ソルバを使う

7. 2 固有周波数と周波数領域の問題

定式化は頻繁に不定の線形問題になり、MG をそのまま使えない。

分離解法もここでは適用できない。

7. 3 スパース線形行列の解法

直接解法

MUMPS, PARDISO, SPOOLES

反復解法

Multilevel 法 : GMS(hp), BiCGSTAB, CG

不完全 LU, iLU(MKL), ILUO(MKL)

注 : MKL は Math Kernel Library

SOR, ヤコビ法 (標準)

SOR ベクトル法 (ベクトル要素 ヘルムホルツ方程式)

SOR ゲージ (ゲージ拘束のない静磁場)

SOR ライン (境界層メッシュ)

SCGS (境界層メッシュ, CFD)

Vanka (鞍点問題)

鞍点問題は線形化した Navier-Stokes 方程式系に生じる。その場合、ヤコビアン行列の対角成分が 0 となる。

他のタイプとして、方程式系が ODE の変数を含む場合に生じる。これらの変数は Vanka アルゴリズムで取り扱わねばならない。SCGS は Vanka を利用するためのオプションである。高次の要素を含むモデルでは SCGS あるいは Vanka スムザーを使わねばならないが、異方性メッシュについてはうまく機能するわけではない。

クリロフ前処理 (ヘルムホルツ方程式)

AMS (補助空間マックスウェル解法, RF)

領域分割 (大規模問題の一般的な分割方法)

領域を解きやすい部分領域に分割し、全体の解は、境界条件として別の部分領域で求めた解をつかうことで各部分領域の解を計算しながら反復的に求めていく。

SAI (スパース近似行列反転)

ハイブリッド前処理

全ての前処理では、この機能を使って複数の前処理の効果を組み合わせることができる。

7. 4 周波数領域と固有値ソルバ

7. 4. 1 周波数領域

単純スイープ (各周波数ごとの計算)

AWE (テーラー展開あるいはパデ近似)

モーダルソルバ 固有関数に基づいた低減系の解法

7. 4. 2 固有値/固有周波数

COMSOL_ReferenceManual.pdf の p.993 に掲載されている。

(1) アーノルド法によるシフト (ARPACK)

一般化固有値問題は以下のようなものである。

$$\begin{aligned}(\lambda - \lambda_0)^2 EU - (\lambda - \lambda_0)DU + KU + N_F \Lambda &= 0 \\ NU &= 0\end{aligned}$$

where the solver evaluates E , D , K , N , and N_F for the solution vector U_0 ; λ denotes the eigenvalue; and λ_0 is the linearization point. If $E = 0$, it is a linear eigenvalue problem; if E is nonzero, it is a quadratic eigenvalue problem. To solve the quadratic eigenvalue problem, COMSOL Multiphysics reformulates it as a linear eigenvalue problem. After constraint handling, it is possible to write the system in the form $Ax = \lambda Bx$.

λ は固有値であり、 λ_0 は線形化ポイントである。 $E=0$ であれば、線形固有値問題になる。 E が 0 でない場合には二次の非線形固有値問題になる。COMSOL は二次の非線形固有値問題を解くために、 $Ax = \lambda Bx$ の形の線形固有値問題に変換している。

境界条件や材料物性が固有値の非線形関数である場合にはより一般的な固有値問題になるが、これらは一連の二次固有値問題を解くことで解決できる。COMSOL は線形化ポイントの周囲で近似的な二次固有値問題を構成することで固有値への一般的な依存性を処理している。線形化ポイントの更新を反復しながら速い収束性を実現している。

シフト量 σ に最も近い固有値を見出すことは、行列

$$(A - \sigma B)^{-1}B.$$

の最大固有値を求めることに等しい。これは ARPACK をベースにソルバを組むことができる。これは不完全リスタートアーノルド法(IRAM)という解法の一つである。

注：ARPACK は ARnoldi PACKage のことであり、大規模固有値解析のために開発されたサブルーチン群であり、対称エルミート系、非対称/非エルミート系の標準固有値問題または一般化固有値問題を解くことができる。

(2) ARPACK を用いた領域サーチ法

この方法は ARPACK に基づいたアルゴリズムを使っており、複素平面内で与えられた、ある十分小さな領域（矩形）の中の全ての固有値を求めることができる。

8. 最適化

8. 1 感度（順方向、随伴）

定常、時間依存について利用

8. 2 最適化（非線形計画 NLP、最小二乗法 LSQ）

勾配法：定常、時間依存に利用

SNPOPT（標準）

Levenberg-Marquardt（LSQ 用）

移動漸近法、MMA（KTH/ Stockholm）

勾配フリー：

座標探索

Nelder-Mead

信頼領域、内挿法： BOBYQA, COBYLA

非線形拘束取り扱い用の拡大ラグランジュ法

9. その他

9. 1 粒子

常微分方程式系による粒子追跡と補助方程式

特別な境界相互作用

乱数による粒子リリース

スティック

バウンス

一般的な場と粒子の相互作用

9. 2 移動メッシュ

ALE

自由境界

変形ジオメトリ

自動リメッシュ

（パラメトリックと時間依存）

9. 3 h 適合（定常、パラメトリック、固有値問題）

L2 ノルム

ゴール志向 (二重重み付残差)

9. 4 h 適合 (時間依存)

サブ時間間隔での固定メッシュ

フィジックスあるいはユーザー制御の誤差関数の利用

9. 5 時間依存のイベント

陰的/陽的

任意の再初期化

9. 6 ゴール志向の誤差評価

定常、周波数領域で利用。

10. Ver. 5.2a 新機能

10. 1 SA-AMG (AMG based smoothed aggregation)

線形弾性解析に効率的。

メモリが効率的。

数100万の自由度をもつ構造計算をノートPCで。

10. 2 領域分割ソルバ

10. 3 時間領域の音響

不連続 Galerkin に基づく陽的ソルバ。メモリ効率。

以上

