

# オープンソフトウェアを用いた Python による FEM 解析

村山 健一\*

(平成 30 年 10 月 31 日受理)

## The FEM Analysis by Python using Open Software

Kenichi MURAYAMA\*

In recent years, with the development of numerical analysis technology, numerical simulation using Finite Element Method (FEM) software has been carried out in various fields. FEM, which is most widely used in numerical analysis software, can be used not only for commercial software but also for free of charge.

In this report, FEM analysis was conducted using an open software without usage restriction, and its computation speed and calculation accuracy were compared with theoretical values and those of other software to confirm its usefulness.

Key word: FEM, CAE, Netgen, Ngsolve, Ngspy, Numpy, Python, Scipy

### 1. まえがき

FEM (有限要素法 : Finite Element Method) による数値解析は航空工学や構造工学の分野から始まり、NASA で開発された NASTRAN をはじめ多くのソフトウェアが開発され、商用ソフトウェアとして広く使われてきた。そして、最近では機械・構造分野の 3 次元 CAD(Computer Aided Design)には FEM が組み込まれているものも多くなっている。

一方、オープンソフトウェアとして無償で使える FEM ソフトウェアは、国外の大学や研究機関を中心に開発されたものが公開され、有償のソフトウェアに匹敵する性能を有するものも現れている。中には、かつては商用として使われていたものがオープンソフトウェアとして公開されたものもある。そして、これらのソフトウェアの活用により CAE(Computer Aided Engineering)技術の普及を図るために OpenCAE 学会が設立され、活発な活動をしている。また、近年、数値解析の分野では Python によって様々なプログラムが作られている。本報告では、Windows 上で Python による FEM 解析を行う場合、最も計算時間を要するマトリックス計算の処理速度を確かめるとともに、Python 上で利用できるオープンソフトウェア Netgen<sup>(1)</sup>/Ngsolve<sup>(2)</sup>/Ngspy<sup>(3)</sup>による FEM 解析を他のソフトウェアと比較し、その有用性を検討する。

### 2. FEM オープンソフトウェアの現状

多くの利用実績があり、高く評価されている本格的な FEM オープンソフトウェアを表

---

\* 教育センター Education Center

1 に示す。この中で **CodeAster** はフランスの電力企業が開発したもので、実績のあることから近年注目され、利用が急速に広まっている。なお、このソフトウェアはメッシュ分割などプリプロセス機能と結果表示のポストプロセス機能がないため、それらのソフトウェアが必要となっている。**Salome-Meca** はプリプロセッサ **SALOME** とポストプロセッサ **Paraview** が **CodeAster** と一体で動作するソフトウェアで、現在は **Windows** 上でも実行可能となっている。これらの中で **Adventure** は国内で開発されたもので、アプリケーションは **Linux** 上で実行されるが、線形構造解析と静磁界解析は **Windows** で動作可能である。このソフトウェアには **CAD** 機能や一部のポスト機能も含まれているが、実用面では他のソフトウェアも必要となっている。そのほか、3 次元 **CAD** として充実してきている **FreeCAD** は **FEM** 機能も搭載されるようになってきており、現在ではメッシュ分割と線形構造解析が可能となっている。

かつて、**FEM** ソフトウェア開発で最も困難な処理はメッシュ分割で、とりわけ 3 次元では容易に開発することはできなかった。しかし、表 2 に示されている **Netgen** と **Gmsh** が公開され、比較的容易に **FEM** 解析できるようになった。この 2 つのソフトウェアで出力されるデータに対応したソルバーは、オープンソフトウェアだけでなく一部の商用ソフトウェアも対応している。また、**Netgen** はソルバー **Ngsolve** と一体で実行できるようになっており、最近では **Python** に対応するようになった。

表 1 FEMソルバー

ソフトウェア	主な機能	特徴	開発元	ライセンス
Code Aster	非線形構造解析	Salome-Mecaで使用	フランス電力公社	GPL
Elmer	線形各種解析	連成解析が可能	フィンランドCSC社	GPL
Culculix	非線形構造解析	Abuqus類似、FreeCADで使用	ドイツ	GPL
Adventure	構造解析・電磁界解析	大規模解析	東京大学	独自
OpenFOAM	流体解析	高汎用性、カスタマイズ可能	イギリス	GPL

表 2 FEMプリ・プロセッサ

ソフトウェア	主な機能	特徴	開発元	ライセンス
Netgen	プリプロセッサ	ソルバーNgsolveと接続	ドイツ	OSL
Gmsh	プリプロセッサ	インターフェースが充実	ベルギー	GPL
SALOME	プリ・ポストプロセッサ	解析モデル作成可能	フランス	GPL
Paraview	ポストプロセッサ	高機能	アメリカ	BSD
PrePoMax	プリ・ポストプロセッサ	Culculix対応、コード非公開	スロベニア	独自

### 3. FEM 解析と性能比較

**Python** プログラムは豊富なライブラリがフリーで利用できるため、さまざまな分野で使われている注目のプログラム言語である。このプログラム言語はインタープリタ型であるにもかかわらず、優れたライブラリが無償で利用できるため科学技術計算における利用が進んでいる。

ここでは最初に、**Python** プログラムで科学技術計算ライブラリによる効果を検討し、

そのあと Netgen によるメッシュ分割と Ngsolve を用いて FEM 解析を行ない、他のソフトウェアとの性能比較を行った。Netgen は 1994 年ドイツで開発され、2003 年からは GPL(GNU General Public License)のオープンソースソフトウェアとなっている。また、計算速度と計算値を比較するために、商用ソフトウェアの Lisa(Ver 8)を用いた。このソフトウェアはカナダで開発されたもので、構造解析をはじめ熱、振動、電界、磁界解析ができるが、線形対応で 32 ビット版であるため、解析規模が限られるものである。なお、プログラムの実行環境は以下のとおりである。

PC : Dell Latitude E6330 , i5-3320M , 2.60GHz, メモリ 8.0Gbyte

OS : Windows 10 , 64bit Professional

開発環境 : Python3.6、Spyder 3.2.8、Jupyter Notebook

### 3.1 Python プログラムの高速化

Python プログラムで数値計算ライブラリを使った場合の速度を比較するために、2 次元解析プログラムをすべて Python で書かれたものと、解を求めるためのマトリックス計算部分を置き換えた場合について、計算時間を検討する。

FEM をすべて Python で実行する場合、解を求めるマトリックス計算が最も大きな負担である。ここでは、1 次 3 角形要素による 2 次元応力解析をガウスの消去法を用いた Python だけによるプログラムと、解を求めるマトリックス計算だけを科学技術計算用ライブラリ Numpy/Scipy が提供する関数に置き換えた場合の計算時間を比較することとした。なお、FEM 解析は、以下の手順で行った。

(1) Netgen によるメッシュ分割、Excel による境界条件、材料データ作成

(2) Python プログラムフロー

- ・データ入力
- ・要素の系マトリックスへの組み込み
- ・境界条件の処理
- ・マトリックス計算
- ・データ処理および保存

FEM 解析の中核を成す系マトリックスは正方形マトリックスとし、最初にプログラム 1 に示されているガウスの消去法で計算した。その後、Python による FEM プログラムの高速化を図るために、この部分を科学技術計算用ライブラリに置き換えた。Numpy では正方形マトリックスに対応した関数、Scipy ではスパースマトリックスに対応した関数を用いた。

処理時間を比較した結果を表 3 に示す。未知数が小さい場合はマトリックス計算の占める割合は小さいが、いずれの場合も 500 を超えるとほとんどの時

プログラム 1 ガウスの消去法

---

```
def gauss(gk,fp):
    n=len(gk)
    for i in range(n):
        for j in range(i+1,n):
            gk[j][i] /= gk[i][i]
            for k in range(i+1,n):
                gk[j][k] -= gk[j][i]*gk[i][k]
            fp[j] -= gk[j][i]*fp[i]
    for i in range(n-1,0,-1):
        for j in range(i+1,n):
            fp[i] -= gk[i][j]*fp[j]
        fp[i] /= gk[i][i]
    return fp
```

---

間がマトリックス計算となっている。正方形マトリックスの計算では、未知数の 3 乗に近い値に比例して計算時間が増加していることが示されている。実用性の点からスパースマトリックスに対応した関数を用いれば、それ以外の処理はほとんど Python プログラムでも未知数が数万程度の FEM 解析が可能となろう。

表 3 Numpy/Scipyとの速度比較

未知数	297	501	1,896	7,404	59,046
Pythonのみ	7.32 s	354 s	18,432 s	計算不能	計算不能
Numpy(正方matrix)	0.219 s	0.531 s	2.43 s	66.8 s	計算不能
Scipy(スパースmatrix)	0.500 s	0.563 s	1.57 s	7.08 s	104 s

### 3.2 応力解析の比較

ここでは簡単な 3 次元応力モデルについて、Netgen でメッシュ分割されたデータを用いて Ngsolve (Ver 6.2) と商用ソフトウェアの Lisa (Ver 8) で計算時間と最大変位を比較する。解析モデルは、図 1 (a)に示されている寸法で、片方が固定され、もう一方に 5,000N の荷重を与えた。鋼材の弾性係数は 200GPa、ポアソン比 0.3 として 4 面体要素数 46,867、未知数 30,840 の FEM 解析を行った。計算値は表 4 に示すように、Ngsolve と Lisa の結果は、理論値に対して 1~2%の誤差でほぼ一致しているが、処理時間は Ngsolve が 3 倍速い結果となった。なお、Ngsolve は必要なパラメータをスクリプト言語に類似したテキストファイルに記述して実行するものである。

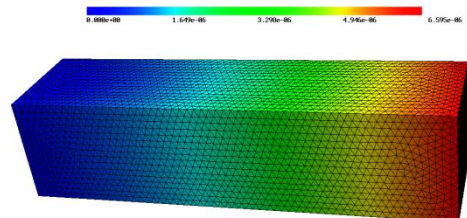
いずれのソフトウェアも C コンパイラで作成されたものであるが、モデル作成、メッシュ分割や結果処理などのプリ・ポスト処理、解析規模、処理速度などいずれもオープンソフトウェアの Netgen/Ngsolve が優れているが操作方法が煩雑で難しい。なお、計算モデルが同一でも計算値に差が出た主な理由は、それぞれのソフトウェアで境界条件の与え方が異なったことによるものである。

表 4 応力解析

ソルバー	要素数	節点数	処理時間	最大変位
Ngsolve	46,867	10,280	3.02 s	$6.595 \times 10^{-6}$
Lisa	46,867	10,280	9.0 s	$6.496 \times 10^{-6}$
理論値	—	—	—	$6.400 \times 10^{-6}$



(a) 解析モデル



(b) 計算結果 (変位)

図 1 応力解析

### 3.3 静電界解析の比較

3次元静電界解析の比較のために、図2(a)に示されているような複数の誘電体内の電位解析を行う。図2(a)はNetgenのCSG(Constructive Solid Geometry)モデル作成機能により作成・表示されたものである。Netgenは、外部よりCADで作成された3次元データを入力してメッシュ分割、境界条件の設定などを行うことができるが、独自のCSGモデルを簡単なテキストデータで作成し、それを透過表示することができる。また、複数材料のメッシュ分割では、最初に材料間の面を分割し、その後、内部立体の分割を行うため、複数材料間で不適合メッシュ分割が行われない特徴がある。これは、複数の誘電体や磁性体で構成され、空気も解析対象になる電界・磁界解析では重要なことである。

ここでは、比誘電率が1と10の2種類で構成される4m角の誘電体中に、半径0.3m、中心間隔1.0mで配置された球状電極の電位解析を行った。図2(b)は電極の一方に1.0V、他方に-1.0Vの電圧を印加した場合の電位分布図である。処理時間は表5に示すようにLisaに比べNgsolveがかなり速い結果となった。さらに、Python上で実行されるライブラリNgspy<sup>(4)</sup>ではさらに高速となった。これは、Pythonの科学技術計算用ライブラリを組み込んでいることによるものと考えられる。なお、計算精度を比較するためにいずれも比誘電率が1の場合について静電容量を求めた。その結果、表5に示されているように、10%前後の誤差となった。この理由は、電極付近の要素サイズが大きく影響しており、全体の要素数が同程度でもメッシュ分割を工夫することにより計算精度は大きく改善されることを確認した。また、その誤差が正と負になる理由は、それぞれのソフトウェアが電極への電荷の与え方が異なるためであった。

表5 静電界解析

ソルバー	要素数	節点数	処理時間	求解計算時間	静電容量
Ngspy	27,997	5,651	時間非表示	0.00840 s	54.90 pF
Ngsolve	27,997	5,651	0.209 s	0.0183 s	53.32 pF
Lisa	27,997	5,651	4.0 s	2.0 s	43.89 pF
理論値	—	—	—	—	47.55 pF

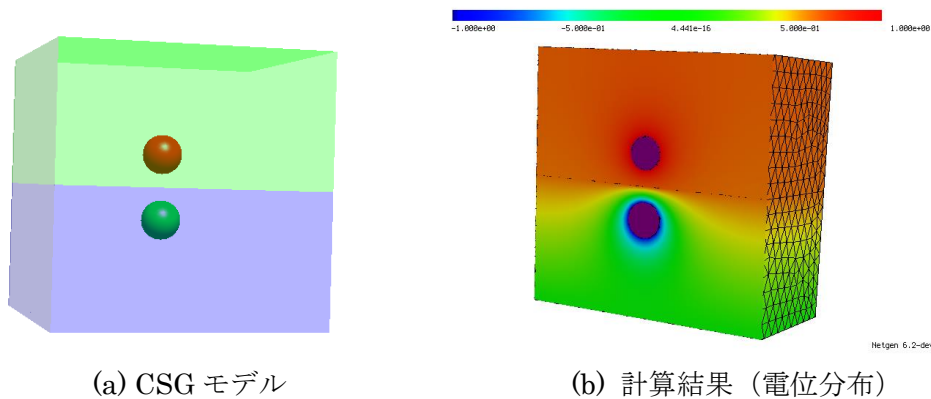


図2 静電解析



### 3.4 静磁界解析

磁界解析は磁性体だけでなく空気を含む周囲も解析対象となるため、とくに 3 次元解析ではモデルの作成が難しくなるが、Netgen を用いれば簡単な 3 次元モデルは容易に作成できる。また、3 次元磁界解析ができるオープンソフトウェアは少なく、商用の Lisa でも解析できない。そこで、ここでは図 3(a)に示されているコイルについて、Ngsolve と 2 次元・軸対称磁界解析のオープンソフトウェア FEMM<sup>(5)</sup>による比較を行うこととした。解析モデルは、磁性体板の近くに直径 1.0m、長さ 3.0m の鉄心に巻かれたコイルに 1,000AT の電流を流したものとした。なお、磁性体板と鉄心の比透磁率は 1,000 とし、空間とコイルは 1 とした。Ngsolve による 3 次元では 4 面体要素数 82,338、節点数 15,899 とし、FEMM では 3 角形要素数 4,231、節点数 8,176 とした。解析の結果は表 6 に示されているように、Ngspy と FEMM は最大磁束密度でよく一致しているが、Ngsolve では結果に差がみられる。そのため、Elmer や Adventure など他のソフトウェアとの比較が必要であろう。

ここで使用した 2 次元・軸対称磁界解析ソフトウェア FEMM は、アメリカの David Meeker らによって開発され、広く知られているものである。現在、バージョンは 4.2 で、スクリプト言語が組み込まれているため、パラメータを変えながら解析を行うことができる。また、Python 上でも実行できることから、解析結果を Python ライブラリで活用することができる。ただ、3 次元解析に対応していないため、利用範囲は限られる。

表6 静磁界解析

ソルバー	解析モデル	要素数	節点数	処理時間	求解計算時間	最大磁束密度
Ngspy	3 次元	82,338	15,899	時間非表示	時間非表示	$3.992 \times 10^{-3}$ T
Ngsolve	3 次元	82,338	15,899	24.66 s	1.19 s	$3.445 \times 10^{-3}$ T
FEMM	軸対象	4,231	8,176	2.0 s	時間非表示	$3.982 \times 10^{-3}$ T

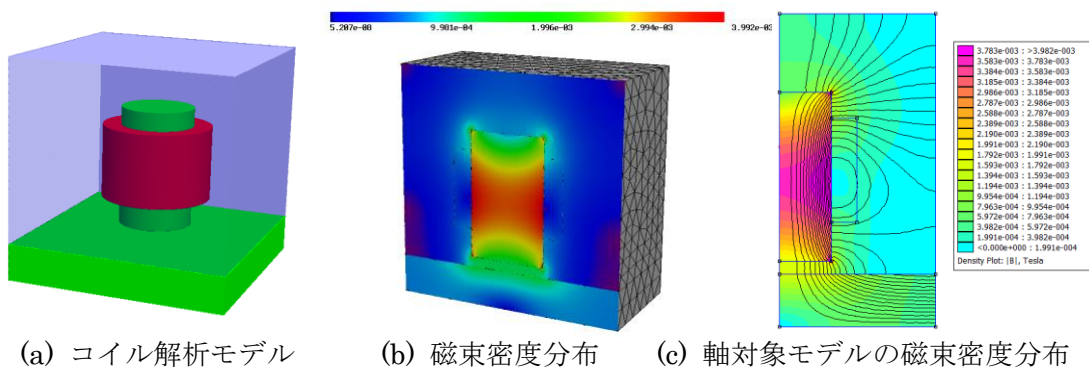


図 3 電磁コイルの静磁界解析

### 4. あとがき

数値シミュレーション技術は、FEM ソフトウェアの開発とオープンソフトウェアの普及により、技術者にとって身近なものとなりつつある。本報告では、数値解析を専門とし

ない技術者の FEM 解析を支援するため、Windows 上で利用できる FEM オープンソフトウェアに限って、理論値や商用ソフトウェアと処理時間や計算値、取り扱い方法の比較を行った。表 7 はその検討結果である。FEM ソフトウェアの使用は、商用であっても取り扱いが難しく、原理を理解するとともに、かなりの経験が必要となっている。さらに、オープンソフトウェアでは、操作方法や関数の使い方の情報が少ないため、優れた性能を有していても利用者は限られ、多様な業務に携わる技術者にとって使いこなすことは困難な状況である。一方、Python による FEM プログラムは開発が容易であるため、技術者独自のプログラムを作りやすい。FEnics<sup>(6)</sup>などオープンソース開発環境を利用すれば効率よく独自の FEM プログラムが作れると思われる。

今後は、さらに活用範囲を広げるために有用な FEM オープンソフトウェアを検討し、県内の技術分野における数値シミュレーション技術の普及・発展に寄与したいと考える。

表 7 オープンソフトウェアの検討結果

機 能	ソフトウェア	比較結果
解析モデル作成	FreeCAD	商用 3 次元 CAD とほぼ同等
	Netgen	CSG モデル作成機能は有効
メッシュ分割	Netgen	メッシュ分割は商用と同等
ソルバー	Ngsolve	高速で大規模データ解析可能
	Ngspy	取り扱いが難しく、説明書が不十分
結果表示	Ngsolve	操作は簡単であるが低機能 他のオープンソフトウェアの利用が必要

#### 参考文献

- (1) Joachim Schoberl, NETGEN - 4.X, January 24, 2009,  
<http://netgen-mesher.sourceforge.net/docs/ng4.pdf>
- (2) Joachim Schoberl, NGSolve – 4.4, April 27, 2018
- (3) Noam Arnold, NETGEN/NGSolve Manual, March 8, 2013
- (4) NGS-Py Finite Element Tool, <https://ngsolve.org/>
- (5) David Meeker, FEMM 4.2 Magnetostatic Tutorial, December 14, 2013,  
<http://www.femm.info/wiki/MagneticTutorial>
- (6) Hans Petter langtangen-Anders Logg, Solving PDEs in Python, The FEniCS Tutorial, January 2016, SpringerOpen