

ランダム変形法とモーメント法によるアンテナの最適設計

大賀 明夫

株式会社 EEM

E-mail: oga@e-em.co.jp

あらまし アンテナの電気的特性を最適設計するために、メッシュ形状の線分の ON/OFF を調整する方法を考える。アンテナの任意の特性を目的関数で表現する。ランダムに選んだ線分の ON/OFF を反転し、目的関数が良くなる時は採用し、悪くなる時は棄却する。この操作を繰り返すことにより最適形状を得る。計算にはモーメント法を用いる。局所解を避けるために乱数を変えて多数回計算し、MPI で並列化し計算時間を短縮する。

キーワード アンテナ, 最適設計, モーメント法, 乱数, MPI

Optimal design of antennas by random deformation method and method of moments

Akio OGA

EEM Inc.

E-mail: oga@e-em.co.jp

1 まえがき

アンテナの用途が広がるにつれてアンテナには放射パターンを任意に制御したり広帯域化したりするなど多様な特性が要求される。

アンテナの特性を最適化するための方法としては、アンテナ形状を少数のパラメーターで表現し、そのパラメーターを最適化する手法が広く行われている。パラメーターが 1 個または 2 個であればしらみつぶしに探すことは可能であるが、より複雑なアンテナではパラメーターの数が増え、最適形状を設計するにはより高度な探索法が必要になる。

ここでは、以上の方法とは異なり、アンテナ形状を非常に多くのパラメーターで表現し、個々の形状要素を離散的に変形することによって最適化することを考える。この方法は組み合わせ最適化とみなすことができる[1]。本方法は多数回の計算を必要とするので小規模問題を高速に計算することができるモーメント法[2][3]を使用する。

2 計算法

2.1 組み合わせ最適化

文献[1]によると組み合わせ最適化は以下のような作業手順となる。

(1) 系の可能な配置の記述

アンテナ形状としては多様な要求を満たすことができるだけの表現能力を持っていることが必要である。ここでは図 1 のアンテナ形状を考える。無限大のグラウンド板の上にアンテナ面(Lx×Ly)を考え、その面をメッシュ分割し(Nx×Ny)、個々の線分の有無(ON/OFF)を考える。アンテナ面の中心とグラウンド板は給電線で結ばれている(同軸給電)。個々の線分はさらに電流分布を精度よく計算するために分割する(図 1 では 2 分割)。

(2) 配置のランダムな変形法

線分配置からランダムに一つを選び、その位置に線分があれば OFF にし、なければ ON にする。なお、線分位置の総数は $N_s = N_x(N_y + 1) + (N_x + 1)N_y$ である。

(3) 最小化すべき目的関数の設計

目的に応じて各種の目的関数が考えられる．例えば+z方向の利得Gを最大にし，入力インピーダンスZ_{in}をZ₀にするためには以下の式が考えられる．

$$F = w_1 G(\theta=0)[dB] + w_2 |Z_{in} - Z_0| \quad (1)$$

ここで，w₁, w₂は重み係数であり目的に応じて適当に調整する．目的関数Fは小さい方が望ましいとしているので利得を高くするには重み係数は負になる．

(4) 反復計算の制御法

文献[1]では巡回セールスマン問題を解くためにアニーリング法を採用している．アニーリング法では変形によって目的関数が小さくなれば必ず採用し，大きくなる時は大きくなる程度に応じた確率で採用する．こうすることによって局所解から脱出できるとされている．

ここでは，目的関数が小さくなれば採用し，大きくなれば棄却するという単純な方法を用いる．計算手順は図2の通りである．

2.2 計算パラメーターの検討

(1) 反復回数

図3に収束状況の一例を示す．これから必要な反復回数は収束状況に応じて1000~5000程度でよい．

(2) 試行回数

図3の収束状態は一つの初期配置から計算した局所解であり，最適値とは限らない．本方法では初期配置をランダムに変えて繰り返し計算して最適値に近い解を探す．この繰り返し回数を試行回数と呼ぶ．

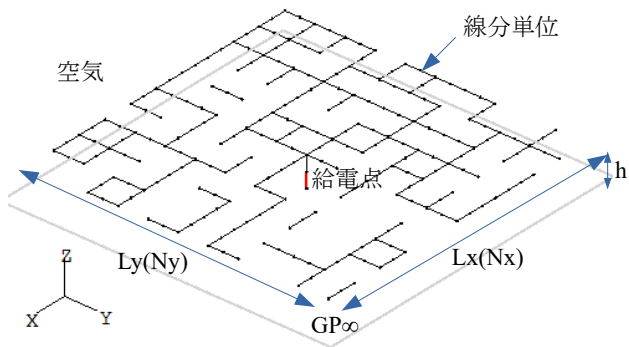


図1 アンテナ形状

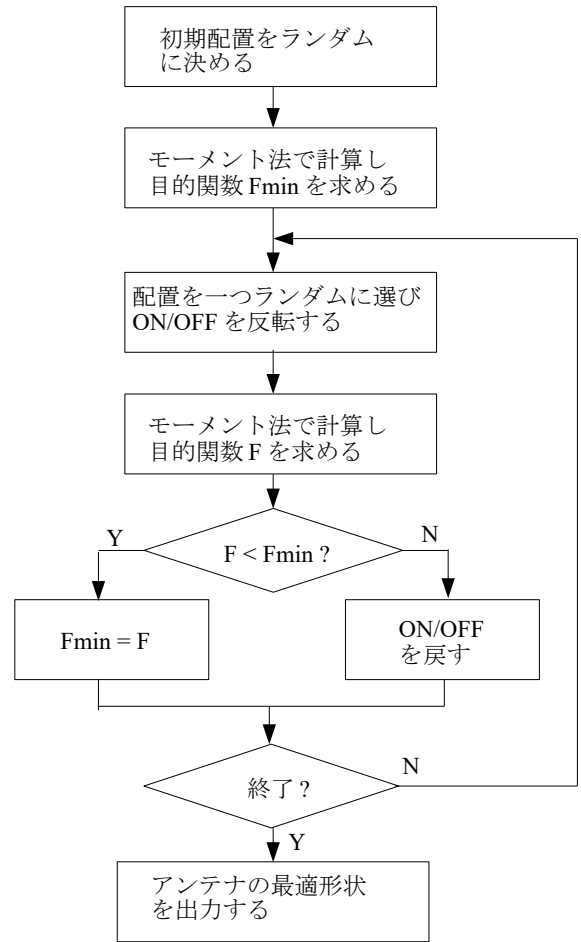


図2 計算手順

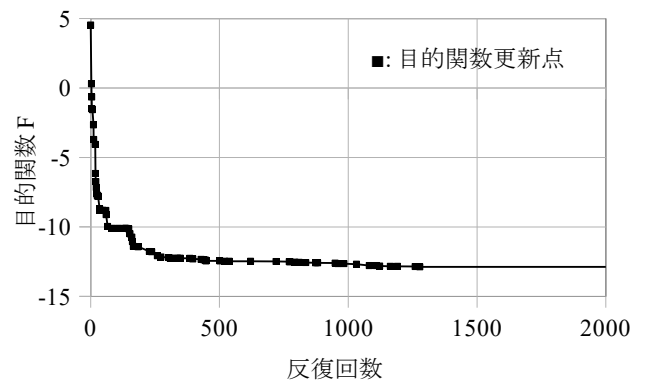


図3 収束状況

Lx=Ly=100mm, Nx=Ny=10, h=10mm, 3GHz
w1=-1, w2=0.05, 初期占有率=0.5

(3) 占有率

総線分配置のうち ON である割合を占有率と呼ぶ．

$$\text{占有率} = \text{ON 線分数} / \text{総線分配置 } N_s \quad (2)$$

図4は初期占有率を0.4~0.6と変えて50回試行したときの各回の目的関数最小値である。計算条件は図3と同じである。初期配置ごとに目的関数最小値が異なるが、10~50回程度試行すればほぼ最適値に近いアンテナ形状が得られることがわかる。また初期占有率=0.5~0.6のとき比較的安定して最適値が得られる。

(4) ハイパーパラメーター

以上からハイパーパラメーター(計算を制御するパラメータ)については以下の目安が得られる。

- ・反復回数=1000~5000
- ・試行回数=10~50
- ・初期占有率=0.5~0.6

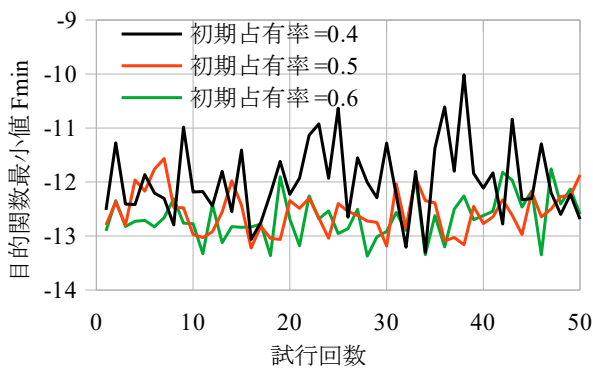


図4 初期占有率と目的関数最小値

2.3 アンテナの高さ

図5はアンテナの高さh(グラウンド板とアンテナ面の距離)を変えたときの目的関数の最小値である。

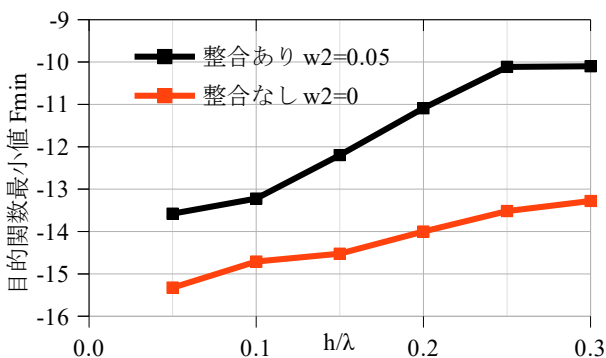


図5 アンテナ高さとの目的関数最小値の関係

$L_x=L_y=100\text{mm}$, $N_x=N_y=10$, 3GHz

$w_1=-1$, 初期占有率=0.5, 反復回数=3000, 試行回数=30

これから、整合の有無によらずアンテナの高さが低いほど目的関数の最小値が小さいことがわかる。 $h=\lambda/4$ のとき利得が最大になるという予想に反している。

2.4 MPIによる並列計算

本方法は多数回の試行が必要になるが、それらの計算は完全に独立なので分散メモリー環境で容易に並列化することができる。ここではMPI(Message Passing Interface)[4]を用いて並列化する。計算手順は以下の通りである。

- (1) 試行回数をプロセス数で分割し、各プロセスに異なる乱数の種を与える、
- (2) 各プロセスが独立に計算し目的関数の最小値を得る。
- (3) 最後に最小の目的関数最小値を持っているプロセスからアンテナ形状を受信し、それを出力する。

3 計算例と計算時間

3.1 計算例

本節では、目的関数を変えて各種のアンテナを最適設計した計算例を示す。共通の計算条件は表1の通りである。なお、目的関数のGの単位はdBi, Z_{in} の単位は Ω である。

表1 共通計算条件

パラメーター	値
$L_x \times L_y$	100 × 100 mm
$N_x \times N_y$ (N_s)	10 × 10 (220)
線分分割数	2
アンテナ高さ h	10 mm
導線半径	0.5 mm
特性インピーダンス Z_0	50 Ω
初期占有率	0.5
反復回数	3000
試行回数	32

(1) 直線偏波, 整合あり

図6に+Z方向のX方向直線偏波の利得を高くし入力インピーダンスの整合を取ったときの結果を示す。左はアンテナの最適形状である。利得は13.6dBiである。一般に特定の偏波成分を最大化すると交差偏波成分は小さくなる。

$$F = -G_\theta(\theta=0, \phi=0) + 0.03|Z_{in} - Z_0|$$

(2) 円偏波, 整合あり

図7に+Z方向の右旋円偏波の利得を高くし入力インピーダンスの整合を取ったときの結果を示す。

$$F = -G_{RHCP}(\theta=0, \phi=0) + 0.03|Z_{in} - Z_0|$$

(3) 2 ビーム, 垂直水平偏波, 整合あり

図8では, 右45度に垂直偏波, 左45度に水平偏波を放射する。

$$F = -\frac{1}{2}\{G_\theta(45, 0) + G_\phi(45, 180)\} + 0.02|Z_{in} - Z_0|$$

(4) 2 ビーム, 左右円偏波, 整合あり

図9では, 右45度に右旋円偏波, 左45度に左旋円偏波を放射する。

$$F = -\frac{1}{2}\{G_{RHCP}(45, 0) + G_{LHCP}(45, 180)\} + 0.02|Z_{in} - Z_0|$$

(5) 2 周波数, XY 直線偏波

図10では, 2.5GHzと3.0GHzの2周波数で共振し, +Z方向に, 2.5GHzではX方向直線偏波, 3.0GHzではY方向直線偏波を放射する。

$$F = -\{G_\theta(f_1) + G_\phi(f_2)\} + 0.03\{|Z_{in}(f_1) - Z_0| + |Z_{in}(f_2) - Z_0|\}$$

(6) 2 周波数, 左右円偏波

図11では, 2.5GHzと3.0GHzの2周波数で共振し, +Z方向に, 2.5GHzでは右旋円偏波, 3.0GHzでは左旋円偏波を放射する。

$$F = -\{G_{RHCP}(f_1) + G_{LHCP}(f_2)\} + 0.03\{|Z_{in}(f_1) - Z_0| + |Z_{in}(f_2) - Z_0|\}$$

(7) 広帯域, 高利得

図12では, 2.5~3.0GHzでS11<-10dBとなり, +Z方向の利得G>10dBiになる。図中の矢印は目的関数を計算した周波数である。参考までにFDTD法(OpenFDTD[5])の計算結果も記す。両者はよく一致している。

$$F = \sum_i \{-G(\theta=0, f_i) + 0.04|Z_{in}(f_i) - Z_0|\}$$

(8) 4 ビーム, 垂直偏波, 整合あり

図13はθ=45度の4方向(±X, ±Y)の垂直偏波成分を最大にしたときの垂直偏波成分の全方向指向性である。4つのビームが確認できる。

$$F = -\frac{1}{4}\{G_\theta(45, 0) + G_\theta(45, 90) + G_\theta(45, 180) + G_\theta(45, 270)\} + 0.02|Z_{in} - Z_0|$$

(9) コニカルビーム, 水平偏波, 整合あり

図14はθ=45度方向の水平偏波のコニカルビームの全方向

指向性である。目的関数の第2項は平坦化のためにある。

$$F = -\frac{1}{N_\phi} \sum_i G(45, \phi_i) + 0.3|G_{max} - G_{min}| + 0.02|Z_{in} - Z_0|$$

$$G_{max} \equiv \max_i G(45, \phi_i), G_{min} \equiv \min_i G(45, \phi_i), \phi_i = 360 \frac{i}{N_\phi}, N_\phi = 36$$

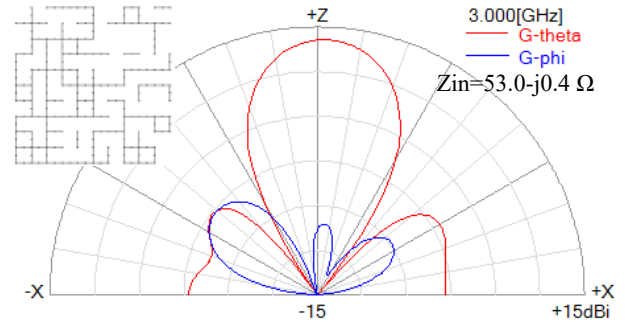


図6 直線偏波, 整合あり

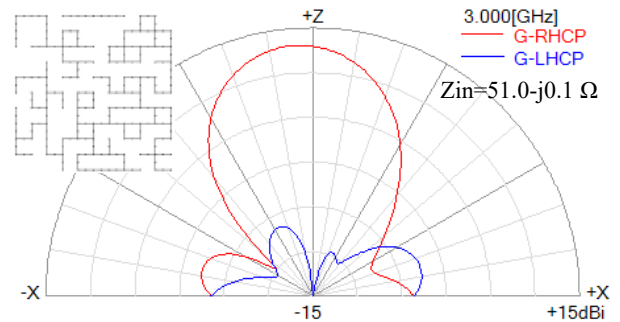


図7 円偏波, 整合あり

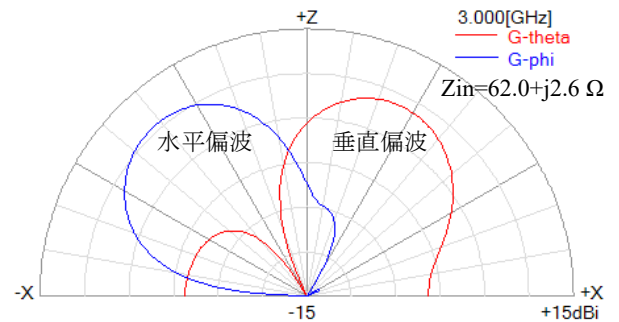


図8 2ビーム, 垂直水平偏波, 整合あり

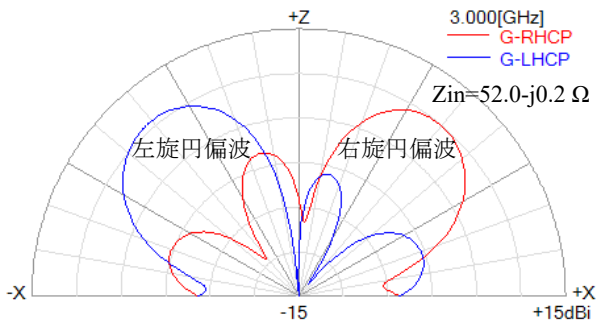


図9 2 ビーム，左右円偏波，整合あり

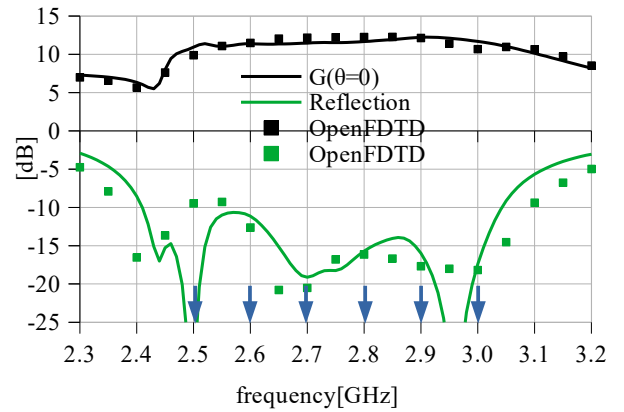


図12 広帯域，高利得，■ ■ : OpenFDTD

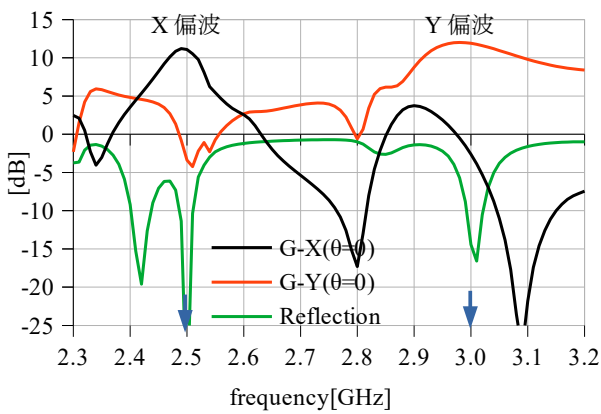


図10 2周波数，下周波数:X偏波，上周波数:Y偏波

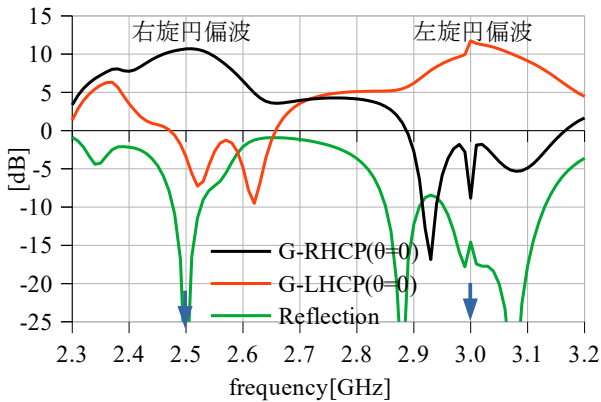


図11 2周波数，下周波数:右円偏波，上周波数:左円偏波

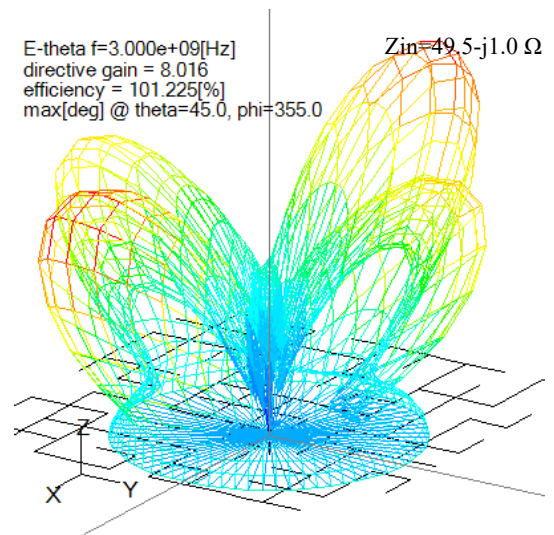


図13 4 ビーム，垂直偏波，整合あり(動径は電界)

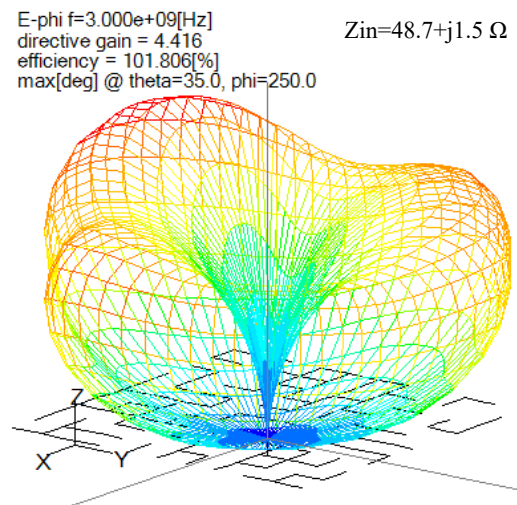


図14 コニカルビーム，水平偏波，整合あり(動径は電界)

3.2 計算時間

モーメント法の線分要素数(=行列の大きさ)は次式で評価される。

$$\text{線分要素数} = N_s \times \text{占有率} \times \text{線分分割数} = 200$$

計算時間は次式で評価される。

$$\text{計算時間} \propto \text{反復回数} \times (\text{試行回数} / \text{プロセス数}) \times \text{周波数の数}$$

なお、OpenMOMは複素数対称密行列を修正コレスキー法[6]で解いておりOpenMP[7]で並列化されている。

図15に計算例(1)に対して1CPUにおいてOpenMPのスレッド数とMPIのプロセス数を変えたときの計算時間を示す。これからMPIの方が線形性が高いことがわかる。すなわち、計算資源が限られているときはモーメント法をマルチスレッドで並列計算するより試行回数をMPIで並列計算する方が効率が良い。なお、本ケースではHT(Hyper Threading)の効果はない。

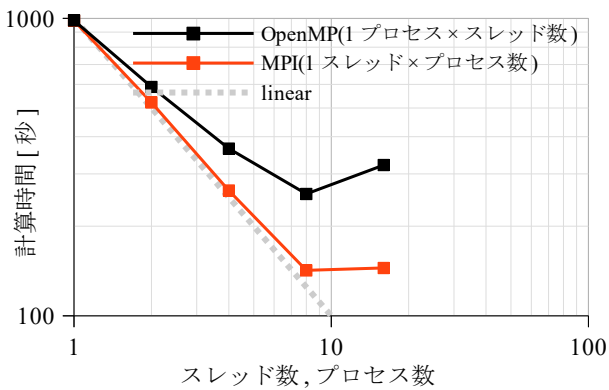


図15 スレッド数, プロセス数と計算時間の関係

Intel Xeon D-1541 (8 コア 16 スレッド×1CPU)

図16に計算例(1)に対してクラスタ環境でノード数を変えたときの計算時間を示す。上から順にFOCUSスパコン[8]のH/X/Wシステムを使用している。これからノード数を増やすとほぼ線形に計算時間が短縮されることがわかる。

4 まとめ

アンテナ形状をランダムに変形し目的関数が小さくなる変形を採用する操作を繰り返すことによって最適形状を得る方法について述べた。初期配置をランダムに変えて多数回試行することによって局所解を避けることができる。

図1のアンテナモデルはアンテナの各種の特性を最適化するだけの表現能力を持っている。

計算にはモーメント法を用いた。最適形状を得るためには10万回のオーダーの計算が必要であるがクラスタ環境においてMPIを用いて並列計算すると計算時間は10秒のオーダーにすることができる(Intel CPU 256コア時)。

本プログラムのソースコードはOpenMOMの付属ツールとして公開している[3]。図1と異なるアンテナモデルについても少しの変更で対応することができる。

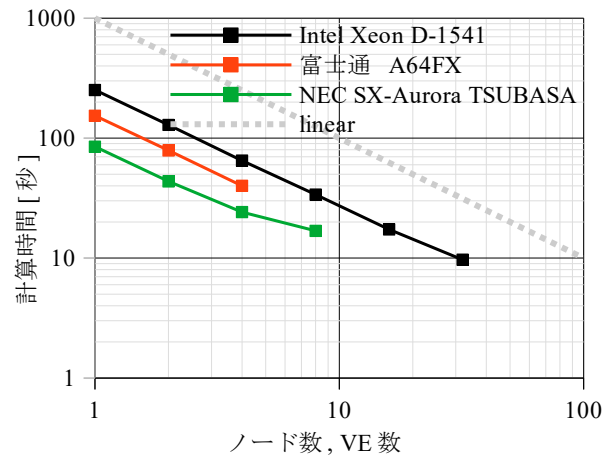


図16 ノード数と計算時間の関係

Intel : 8 スレッド×1 プロセス / ノード

富士通: 12 スレッド×4 プロセス / ノード

NEC : 1 スレッド×8 プロセス / VE (8VEのときは2×4)

文 献

- [1] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, (丹慶勝市他訳), "Numerical Recipes in C, C言語による数値計算のレシピ," 技術評論社, 1993.
- [2] R. F. Harrington, "Field Computation by Moment Methods," Wiley-IEEE Press, 1993.
- [3] OpenMOM, <http://www.e-em.co.jp/OpenMOM/>
- [4] P. パチュコ (秋葉博訳), "MPI並列プログラミング," 培風館, 2001.
- [5] OpenFDTD, <http://www.e-em.co.jp/OpenFDTD/>
- [6] 森正武, "数値解析(第2版)," 共立出版, 2002.
- [7] 菅原清文, "C/C++プログラマーのためのOpenMP並列プログラミング(第2版)," カットシステム, 2012.
- [8] (公財)計算科学振興財団, FOCUSスパコン, <https://www.j-focus.or.jp/>