

# 効果的なサブセクションの制御

Volker Muhhaus \*

石飛 徳昌 †

2009年9月19日

## 概要

この文書では様々な例題に基づいて Sonnet の効果的なサブセクションについて論じます。そしてきめ細かいサブセクションの指定がメモリを節約し、解析時間を短縮する事を示します。

## 目次

1	サブセクションの基本:staircase(階段)と conformal(不定形)のサブセクション
2	どの fill type を使うか?
3	なぜ, staircase が効果的な場合と conformal が効果的な場合があるのか?
4	スパイラルインダクタの例
5	低温焼成セラミックス基板中のダイプレクサの例
6	メモリ使用量やメッシュ数をどうやって調べるか?
7	どうやってサブセクションの密度を制御するか?
8	応用例:マイクロ波フィルター
9	応用例:小型 UWB フィルター

10	応用例:グラウンドシールド付きスパイラルインダクタ	9
11	応用例:多数のヴィアの集合を持つスパイラルインダクタ	11
12	サブセクションを制御するためのダミーの図形	12

## 1 サブセクションの基本:staircase(階段)と conformal(不定形)のサブセクション

1

2

3

3

4

5

5

7

8

Sonnet はユーザーが指定するセルサイズに従って問題を細かい部分に分割して解析します。分割の最小単位をセルと呼び、整数個のセルをまとめた解析単位をサブセクションと呼びます。図 1 には Sonnet が使用する様々なサブセクションを示します。<sup>\*1</sup> サブセクションは staircase つまり階段状または, conformal(不定形) です。staircase サブセクションは長方形で、その辺は Sonnet の解析 box の壁と平行です。また電流の変化は x,y 方向に線形に変化します。

conformal サブセクションは自由で様々な形状になります。conformal サブセクションの形状は導体の形状に沿い、そして電流も staircase の場合より自由で変化に富みます。しかし conformal サブセクションは伝送線路のように、その幅が波長より狭

<sup>\*1</sup> 電磁界シミュレータが問題を分割する時の単位としてメッシュという表現の方が一般的です。Sonnet の文書やメニューにもメッシュという語とサブセクションという語が混在しています。この文書では解析メッシュに相当するサブセクションが複数のセルの集合であるという概念を明確にするためサブセクションという語に統一します。

\* Dr.Muhlhaus Consulting and Software GmbH

† 有限会社ソネット技研

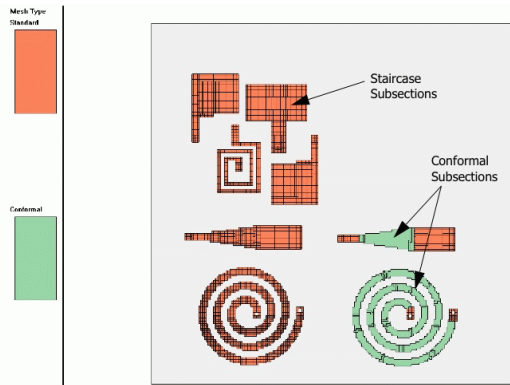


図1 様々なサブセクション

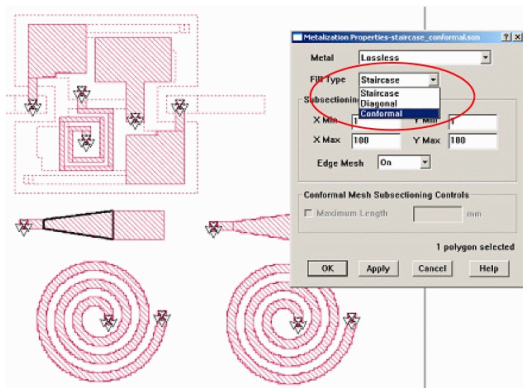


図2 metal properties ダイアログボックス

い場合にしか適用できません。

conformal サブセクションは Sonnet 以外のシミュレータで使われている三角サブセクションに似た印象があります。しかしそれは間違いです。三角サブセクションではそのサブセクション内の電流は、Sonnet の staircase サブセクションと同じく制約されます。conformal サブセクションは以下の例に示すように遥かに強力にフレキシブルです。

デフォルトでは Sonnet は staircase サブセクションを全ての導体図形に対して（特別な via サブセクションを除く）使います。しかし個々の導体図形毎に subsection を設定できます。それは図2の metal properties ダイアログボックスの Fill type で指定します。ただし、もしその導体図形が conformal サブセクションに適さなければ conformal サブセクションの指定は staircase サブセクションに戻され

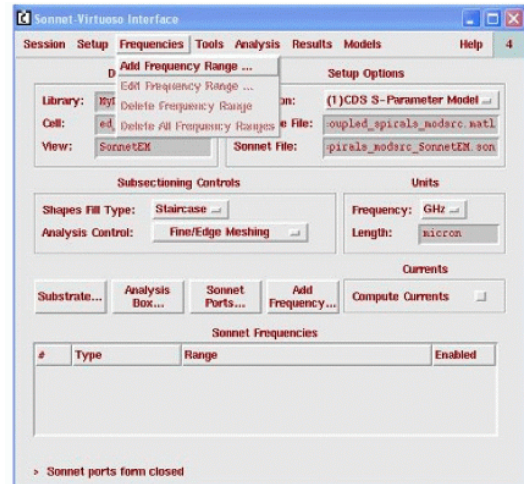


図3 Sonnet Virtuoso Interface

てしまいます。

古いバージョンとの整合性を保つため、Diagonal(対角) という第3のサブセクションが残されています。それは staircase に似ていますが、45° の斜めの線を正しく表現できる方法です。conformal サブセクションが開発される以前に、それは良い選択肢でしたが、現在では conformal サブセクションに置き換えられるべき機能です。

Sonnet を Agilent インターフェースや Cadence インターフェースを通して使用している場合、インターフェースが全ての Fill type を一様に指定してしまいます。個々の導体図形毎のサブセクションを指定するには、そのモデルを Sonnet のプロジェクトエディタで開かなければなりません。(図3)

## 2 どの fill type を使うか？

staircase サブセクションは辺が解析 box の壁に平行な導体図形に対して効果的です。

- 方形のスパイラルインダクタ
- 長方形の導体図形で構成された回路パターン

conformal サブセクションは、解析 box と平行でない斜めの図形や曲った線路に効果的です。

- 円形あるいは八角形のスパイラルインダクタ
- 曲線あるいは幅が徐々に変化する伝送線路

さらに次のような場合は conformal サブセクションを使ってはいけません。(たとえ conformal サブセクションを指定しても Sonnet が無視するかもしれません。)

- 波長に比べて無視できない幅、あるいは細長く無い形状
- 多くの via を持った導体図形

## 2.1 注意

この文書で説明するサブセクションの細かな指定は Sonnet 製品のグレードによって使用できない事があります。

- conformal サブセクション機能は Sonnet Professional にのみ実装されています。
- diagonal サブセクション機能は Sonnet Professional にのみ実装されています。
- 導体図形毎のサブセクション指定は Sonnet Professional にのみ実装されています。
- Cadence インターフェースは Sonnet Professional にのみオプション設定されています。
- Agilent インターフェースは Sonnet Professional にオプション設定されています。
- Agilent インターフェースは Sonnet Lite/LitePlus に実装されています。

## 3 なぜ、staircase が効果的な場合と conformal が効果的な場合があるのか？

Sonnet の解析は二つの工程で行われます。第一に Matrix fill, 第二に Matrix Solve です。Matrix fill 工程では、各サブセクションに単位電流を流した時、他のサブセクションにどのような電圧が生じるかをしらみつぶしに調べてゆきます。そしてその結果を表、つまり Matrix に記録してゆきます。サブセクションの数が  $N$  個あれば、その Matrix は  $N \times N$  の大きさになります。第二の Matrix Solve 工程では、その  $N \times N$  行列の逆行列を計算します。この行列を記録するためのメモリ容量は  $N^2$  に比例します。そして Matrix Solve に必要な時間は概ね  $N^3$  に比例します。従ってサブセクション数  $N$  が

大きい場合に全体の解析時間の中の多くが Matrix Solve の時間になります。

staircase サブセクションでは、各サブセクションの電流の流れは単純なので Matrix Fill の工程は容易で早く終わります。しかし staircase サブセクションの数が多いと Matrix の大きさも大きくなり Matrix Solve に多くの時間がかかります。

このことは円形スパイラルインダクタでは深刻な問題です。円形スパイラルインダクタを staircase サブセクションに分割するとサブセクション数は膨大でメモリ使用量も計算時間も許容できないレベルになります。この問題解決のために conformal サブセクションが導入されました。

conformal サブセクションではサブセクション中の電流に強い自由度を与え、屈曲した導体形状を流れる電流を少ないサブセクションで再現します。これによりサブセクションの数は大幅に減り、Matrix サイズを小さくし計算時間を短縮できます。しかし conformal サブセクションの中を流れる複雑な電流が、他のサブセクションに与える影響を調べるのは困難で、Matrix Fill の時間は staircase サブセクションの場合より長くなります。staircase を選ぶか conformal を選ぶかは、Matrix fill と Matrix Solve の時間のバランスに依存するのです。もし Matrix Solve の時間が支配的であれば、斜めや曲線の導体図形に conformal サブセクションを使う事でサブセクションの数を減らせるかどうかを調べる価値があります。

## 4 スパイラルインダクタの例

この例では、図 4 のスパイラルインダクタに対する conformal サブセクションの効果を示します。解析は Sonnet 12.01 の High Performance Solver を 8 コアの Dell T7400 Dual Xeon5420 で実行しました。

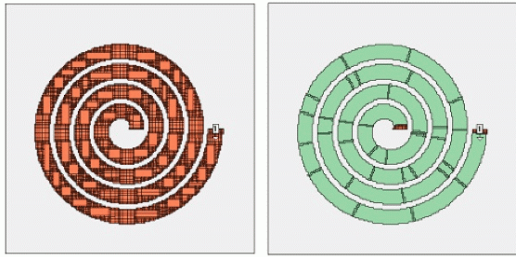


図4 スパイラルインダクタのサブセクション (左:staircase サブセクション 右:conformal サブセクション)

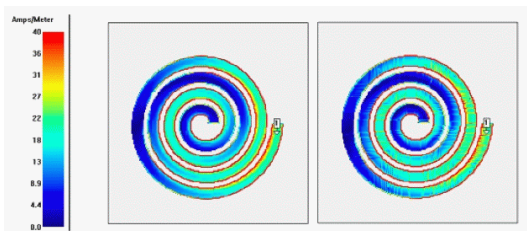


図5 スパイラルインダクタの電流分布 (左:staircase サブセクション 右:conformal サブセクション)

	staircase	conformal
Nos	10612	520
conformal Nos		52k
メモリ使用量	862MB	6MB
Matrix Fill	5 秒	5 秒
Matrix Solve	61 秒	0.03 秒
解析時間	70 秒	6 秒

表のようにこの問題を staircase サブセクションの場合と conformal サブセクションの場合で比較すると、計算時間は1桁以上違います。

図5の電流密度分布で conformal サブセクションを使った解析結果を見ると、微かな乱れがあります。しかし導体の端部の集中や線路に沿った変化等の重要な電流分布が再現されています。各 conformal サブセクション内の電流は精密に解析されています。

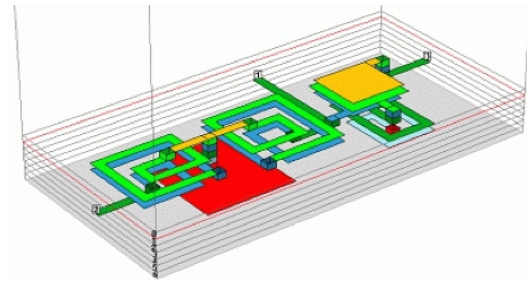


図6 低温焼成セラミック基板に構成したダイプレクサ

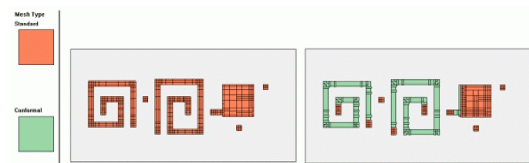


図7 サブセクションの様子 (左:デフォルトの staircase 指定 右:conformal を指定しても一部の導体図形は staircase に分割される)

## 5 低温焼成セラミックス基板中のダイプレクサの例

図6は低温焼成セラミックス基板の中に構成された周波数ダイプレクサで、導体パターンは長方形の導体形状だけで構成されています。それゆえこの問題は staircase サブセクションに適しています。

	staircase	conformal
Nos	3000	2754
conformal Nos		10k
メモリ使用量	71MB	60MB
Matrix Fill	3 秒	6 秒
Matrix Solve	1 秒	1 秒
解析時間	7 秒	10 秒

サブセクションの指定を変えて、解析してみると、staircase サブセクションの方が conformal サブセクションより早い事がはっきりします。

図7はサブセクションの様子です。この例では conformal サブセクションのほうが解析に長い時間がかかっています。Matrix Fill の時間が長い事です。また via の周辺や大きなキャパシタ電極

\*2 Nos: Number of Subsections サブセクションの数

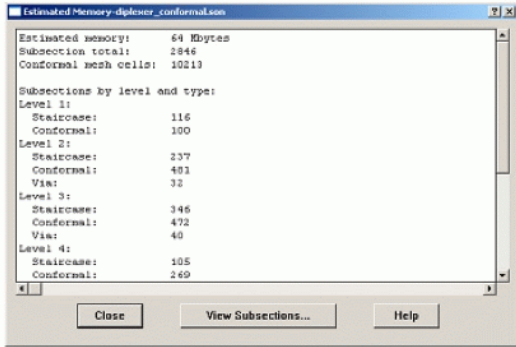


図 8 Estimate Memory:メモリの見積

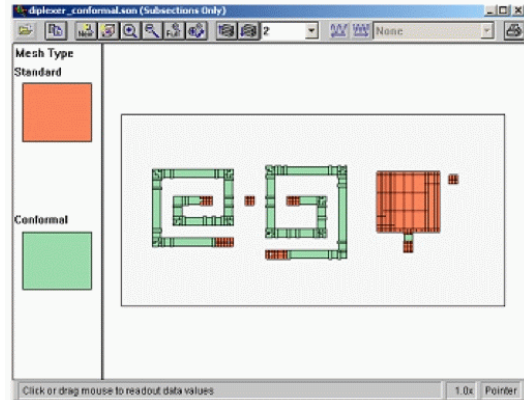


図 9 サブセクションの種類の表示

では解析エンジンが Conformal サブセクションを staircase サブセクションに切り替えている様子がわかります。

このモデルに conformal サブセクションを使う事はできますが、staircase サブセクションの方が効果的です。

## 6 メモリ使用量やメッシュ数をどうやって調べるか？

Sonnet Project Editor では [Analysis]-[Estimate Memory], Agilent や Cadence インターフェースを通じて Sonnet を使っている場合は [Sonnet]-[Analysis]-[Estimate Memory] を選んでください。

図 8 のように必要なメモリ、サブセクションの数などが表示されます。またサブセクションの数は層毎に表示されます。

ここからさらに [View Subsections] ボタンをクリックすると、サブセクションの様子がグラフィカルに表示されます。さらに上下のカーソルキーで各レイアを移動することができます。(図 9)

## 7 どうやってサブセクションの密度を制御するか？

二つの方法で staircase のサブセクションの密度を制御できます。

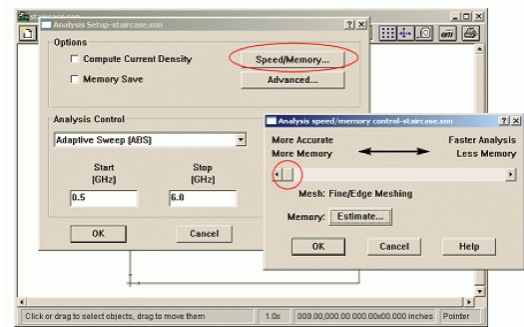


図 10 speed/memory スライドバー

### 7.1 speed/memory スライドバー

デフォルトでは、via や、不連続や、導体図形が重なり合った部分そして図形の端部（端部の電流集中を再現するため）に、指定のセルサイズで許される限り小さなサブセクションが配置されます。このアルゴリズムは、モデル全体と各図形に対してそれぞれ制御できます。モデル全体に対する設定は [Analysis]-[Setup] メニューの [Speed/Memory] ボタンで現れます。(図 10)

このダイアログボックスでは、メッシュの密度を 3 段階に制御できます。図 11 はその一例で、セルサイズは一定のままメモリ使用量は 93MB,45MB,22MB に変化します。

デフォルトのサブセクションでは、図 12 のように導体図形の角に 1 セル、そして端部に沿って 1 セル幅のサブセクションが置かれます。導体図形の内側に進むに従ってサブセクションは 2 倍に大きくな

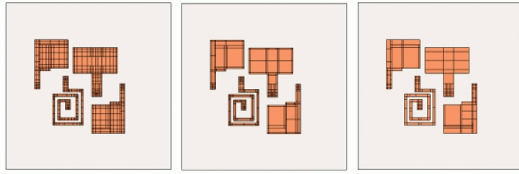


図 11 スライダーの位置とサブセクション

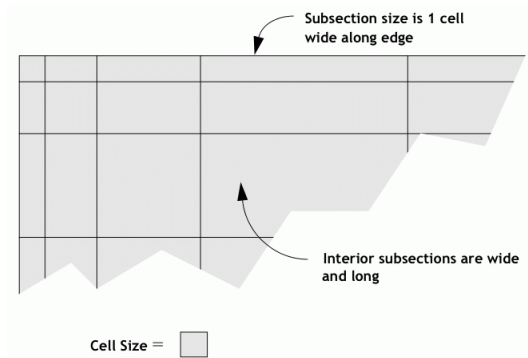


図 12 スライダー左端のサブセクション

ります。しかしサブセクションは最高解析周波数での  $1/20$  波長<sup>\*3</sup>より大きくなりません。

スライダーはサブセクションの二つの項目を制御します。<sup>\*4</sup>

- 導体図形端部の細かいサブセクションを使うか使わないか？
- 導体内部のサブセクションを大きくする比率

Speed/memory スライダーはデフォルトで左端の”More Accurate,More Memory”です。導体図形端部に細かいサブセクションが配置され、導体内部に進むに従って 2 倍ずつサブセクションは大きくなります。(図 12)

Speed/memory スライダーが中央の時、導体図形端部に細かいサブセクションが配置されますが、導体内部にはいきなり大きなサブセクションが配置されます。(図 13)

<sup>\*3</sup> 最高解析周波数での波長は、全体に最も大きな誘電体が満たされたと仮定した長さが使われます。Professional 版では、この制限は [Analysis]-[Advanced Subsectioning] で自在に変更できます。

<sup>\*4</sup> Speed/Memory スライダーは conformal サブセクションには影響しません。

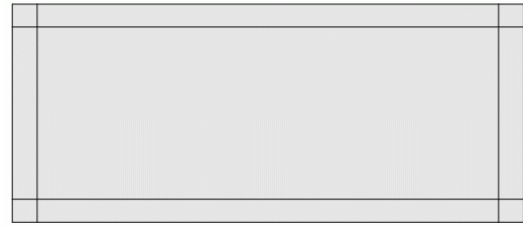


図 13 スライダー中央のサブセクション

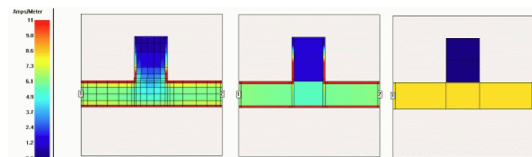


図 14 スライダーの位置と電流密度解析結果

Speed/memory スライダーを右端の”Faster Analysis,Less Memory”にすると、導体図形全体に大きなサブセクションが置かれ、導体図形端部の電流集中は再現されません。この事は解析結果に無視できない誤差を生じます。

図 14 は Speed/Memory スライダーの設定による電流密度の解析結果の例です。黒い細い線はサブセクションを現します。サブセクションは全て staircase サブセクションです。最高解析周波数は 100Mhz で、サブセクションサイズは  $1/20$  波長未満に制限されています。

スライダーが左端の図では、多くのサブセクションが置かれ、電流密度分布の重要な現象、端部への電流集中、線路からスタブへの電流が流れ込む様子が観察できます。中央では、詳細な現象は失われますが、全体としてよく似た電流密度分布が再現されます。右端ではサブセクションは大きく視覚的情報は殆ど得られません。

現実の多くの問題で、Speed/Memory スライダーは中央に設定すれば Speed と Memory とそして精度に関して満足な結果が得られます。左端の精密なサブセクションは、論文の図などで、電流密度分布そのものを視覚的に示したい場合に限るべきでしょう。また、右端はラフに早く解析したい場合や、メモリの使用量が限られている場合に使用して

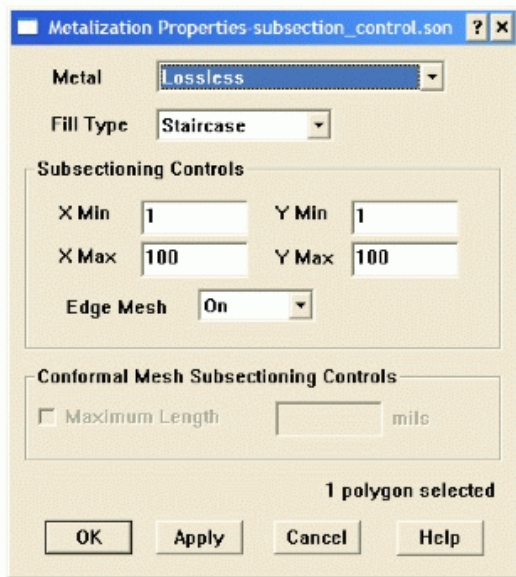


図 15 Ssubsection Controls ダイアログボックス

ください。<sup>\*5</sup>

## 7.2 図形毎のサブセクション密度: $x_{min}/y_{min}$ と $x_{max}/y_{max}$

上記でモデル全体に対するサブセクションが Speed/Memory スライドバーで制御できる事を説明しました。ここではさらに各導体図形それぞれの staircase サブセクションの密度を制御する方法を紹介します。

デフォルトでは導体図形の角には  $1 \times 1$  セルサイズのサブセクションが、そして導体図形の端部に沿って幅 1 セルサイズのサブセクションが置かれます。導体図形内部のサブセクションは最高解析周波数での  $1/20$  波長に達するまで 2 倍毎に拡大されます。

導体図形をダブルクリックすると "Subsection Controls" ダイアログが開きます。(図 15) このダイアログでは、最大と最小のサブセクションサイズ、そして端部の細かいサブセクション (Edge Mesh)

<sup>\*5</sup> セルサイズを異常に小さく設定して、Speed/Memory スライドバーでサブセクションやメモリ使用量を制御してはいけません。その場合、多数のセルを組み合わせるサブセクションを生成する Matrix Fill 工程に多量のコンピュータリソースを使ってしまい、計算時間の節約にならないでしょう。

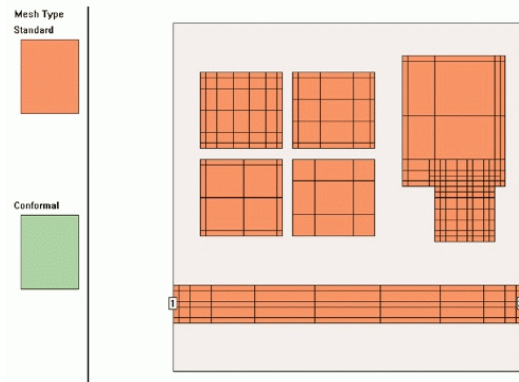


図 16  $x_{min}/y_{min}$  とサブセクション

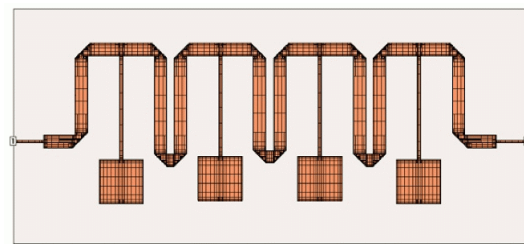


図 17 超伝導マイクロストリップフィルターのデフォルトサブセクション

を使うかどうかを設定できます。

$x_{min}/y_{min}$  を設定すると、モデル全体に対する設定よりも大きなサブセクションを設定することができます。この機能を使うためには Speed/Memory スライドバーを左端の "More Accurate, More Memory" に設定しておいて、その後に必要な導体図形に対して大きなサブセクションを設定します。

図 16 は様々な  $x_{min}/y_{min}$  の設定によるサブセクションの様子です。左上の 4 つの四角のうち左上の導体図形はデフォルトの設定 ( $x_{min} = 1, y_{min} = 1$ ) を中上の導体図形は  $x_{min} = 2, y_{min} = 2$  です。

左中の導体図形は  $x_{min} = 4, y_{min} = 4$  で、中々の導体図形はさらに  $edgemes = off$  の場合です。

## 8 応用例: マイクロ波フィルター

図 17 は超伝導導体を使ったマイクロストリップフィルターの例です。給電線路の幅は  $160\mu m$  で、

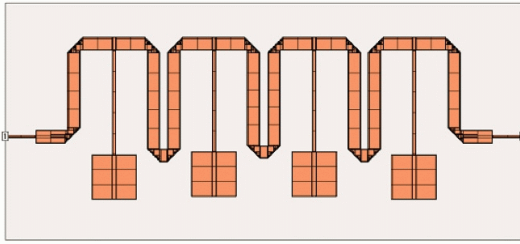


図 18 スライドバーを中央にしたサブセクション

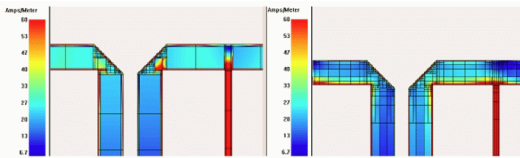


図 19 サブセクションによる電流分布の違い

パッチに接続する細い線路は 240um です。精度を上げるためにセルサイズは 40um に設定してあります。それゆえ、最も細い 160um の線路ですら、線路幅に 4 つのセルが含まれます。

このように細かいセルを設定すると、デフォルトではサブセクションが高密度に配置されます。デフォルトでは 8644 のサブセクションが配置され、メモリ使用量は 311MB でした。解析時間は一周波数あたり 35 秒でした。(Matrix Fill に 18 秒, Matrix Solve に 16 秒, Dell T7400 Dual Xeon 5420 と Sonnet 12.01)

Speed/Memory スライドバーを中央に設定すれば、導体内部のサブセクションを大きくしますが、導体端部には細かいサブセクションが置かれます。(図 18) サブセクションの数は 3769 で 43% に減り、メモリ使用量も 81MB で 26% になります。解析時間は一周波数あたり 9 秒で 25% に短縮されます。(Matrix Fill に 7 秒, Matrix Solve に 1 秒)

図 19 のようにサブセクションを減らすと明らかに電流密度の再現性は悪くなります。この事はモデルと、その目的によって重要かもしれませんが、そうでないかもしれません。

二つの解析結果を重ね描きしてみると両者は非常によく一致することがわかります。(図 20) それゆえ通常は、Speed/Memory スライドバーを中央に

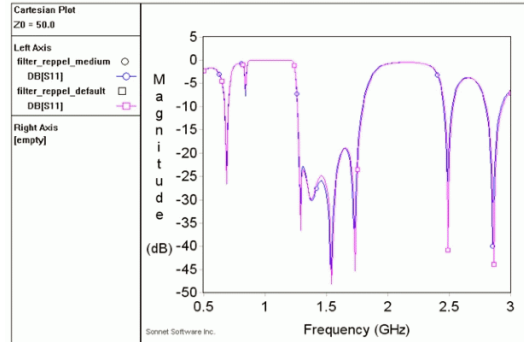


図 20 サブセクションによる周波数特性の違い

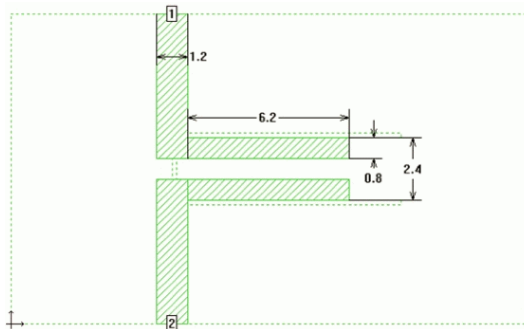


図 21 UWB フィルターの導体パターン (マイクロストリップ側)

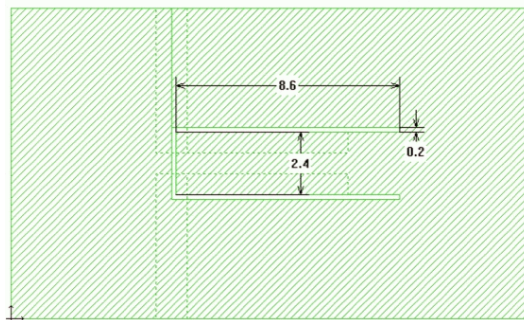


図 22 UWB フィルターの導体パターン (CPW 側)

設定する事を御勧めします。設計の最終段階で細かいサブセクションを使ってチェックする事は良い事でしょう。

## 9 応用例:小型 UWB フィルター

このモデルは電氣的にとっても大きなモデルです。1/4 波長の CPW 共振器が、入出力のマイクロストリップ線路と結合しています。CPW 側とマイクロ



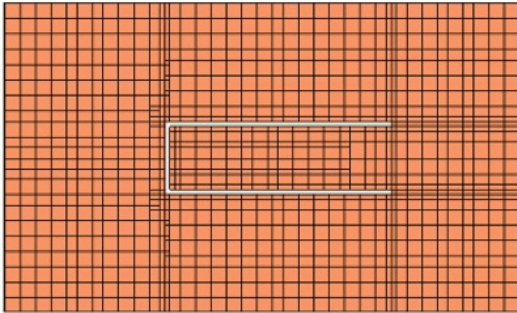


図 23 広いグラウンド導体上のセクション (セルサイズ 0.2)

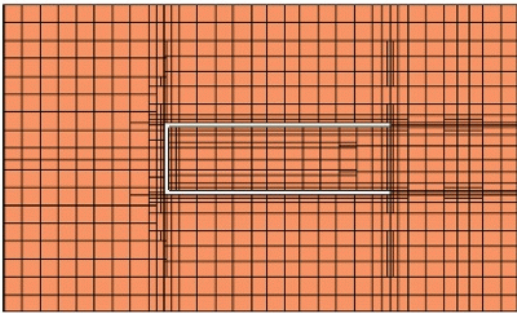


図 24 広いグラウンド導体上のセクション (セルサイズ 0.1)

ストリップ側の導体パターン図をそれぞれ図 22, 図 21 に示します。

Sonnet は CPW の広いグラウンド導体に多くのサブセクションを置いて解析しなければなりません。それゆえ一般に CPW 構造は Sonnet にとって不利な問題です。幸いにスロットの幅は 0.2mm なので、セルサイズを 0.2mm に設定できます。

CPW のグラウンド導体は 1 波長以上の大きさなので、多くの 1/20 波長のサブセクションが配置されます。デフォルトの設定ではサブセクションは 2298 個、メモリ使用量は 45MB、解析時間は 3 秒 (Matrix Fill に 1 秒, Matrix Solve に 1 秒, Port calibration に 1 秒) でした。(図 23)

この問題では Speed/Memory スライドバーをデフォルトの左端から中央にしても殆ど効果がありません。導体図形内部のサブセクションはどうせ 1/20 波長に制限されているからです。

ここでセルサイズを 0.2mm から 0.1mm に小さ

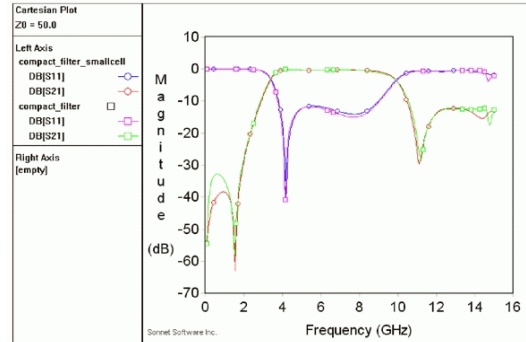


図 25 セルサイズによる解析結果の違い

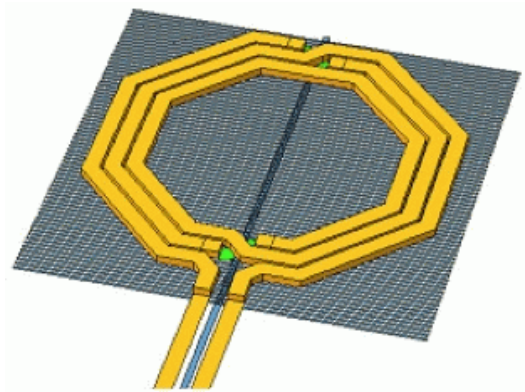


図 26 グラウンドシールド付きのスパイラルインダクタ

くします。図 24 のように、セルサイズを細かくしても導体内部のサブセクションはほとんど変化しません。しかしスロットの周辺には細かなサブセクションが追加され、精度は向上します。サブセクションは 2500 個、メモリ使用量は 55MB、計算時間は 6 秒でした。セルサイズを半分にした結果それぞれ +9%, +22%, +100% 変化しました。図 25 は解析結果です。

## 10 応用例: グラウンドシールド付きスパイラルインダクタ

グラウンドシールド付きのスパイラルインダクタの例を示します。(図 26)

このインダクタの巻線自体は conformal サブセクションに分割され、グラウンドシールドには staircase サブセクションが置かれます。しかしここでは、層間のサブセクションを相互に影響させる仕組みが問

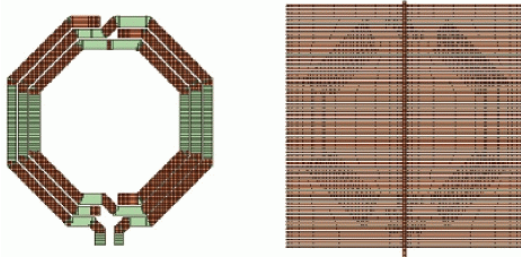


図 27 グランドシールド付きのスパイラルインダクタのデフォルトサブセクション

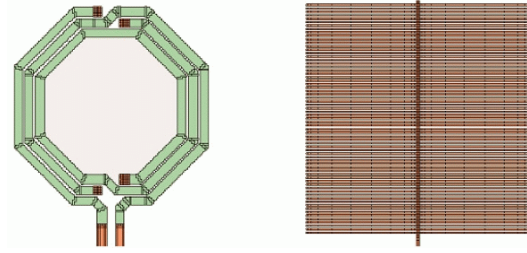


図 29 Polygon Edge Checking 0 の場合のサブセクション

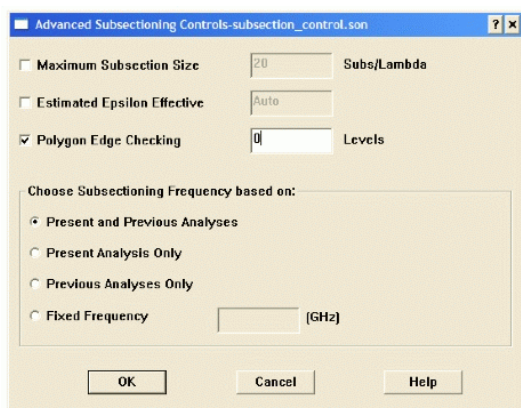


図 28 Advanced Subsectioning ダイアログボックス

題を引き起こすことがあります。この仕組みは各層のサブセクションの境界を、層間のキャパシタを高精度に解析できるように設定します。この仕組みにより、このスパイラルインダクタの例では、インダクタの斜めの導体に沿って膨大な数の staircase サブセクションがグランドシールド導体に配置されてしまいます。必要なメモリ使用量は 2.7GB にも達してしまいます。(図 27) より効果的なサブセクションを配置するために工夫が必要です。

最初にすべき事は、別の層のサブセクションからの影響を取り除く設定です。[Analysis]-[Advanced Subsectioning] でダイアログボックスが開きます。"Polygon Edge Checking" のデフォルト値は 1 です。これは隣接する層のサブセクションのエッジを検出して、その層のサブセクションを決めるということです。これは多くの場合に適切な設定です。多くの層が影響し合っているような場合にはこの値を増やすべき場合もあります。(例えば、多層

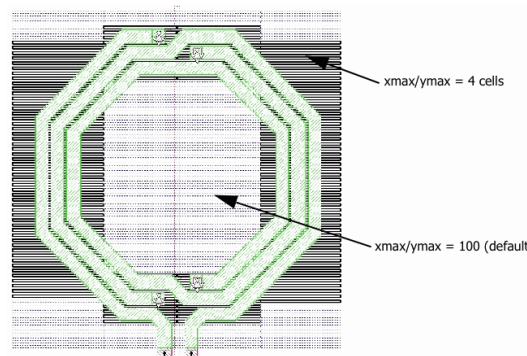


図 30 グランドシールド導体の分割とサブセクションの個別指定

構造の MIM キャパシタ) しかしこのスパイラルインダクタの場合は、この"Polygon Edge Checking" は 0 に設定し、手で適切なサブセクションを設定した方がよいです。(図 28)

"Polygon Edge Checking" を 0 に設定し、他にも変更しないとメモリ使用量は 135MB になり、サブセクションは図 29 のようになります。

インダクタの巻線層とグランドシールド導体の間のキャパシタの精度を改良するべく、サブセクションを手動で変更しましょう。インダクタの巻線とグランドシールド導体が対向している領域ではより小さなサブセクションが好ましいです。そこで [Edit]-[Divide Polygons] を使って、グランドシールド導体を分割し、図 30 のように部分的に  $x_{max}=4, y_{max}=4$  を設定しましょう。これによりサブセクションは各方向に 4 セルより大きくなりません。この設定は各導体図形のプロパティで設定できますし、複数の導体図形を選択して一気に設定することもできます。

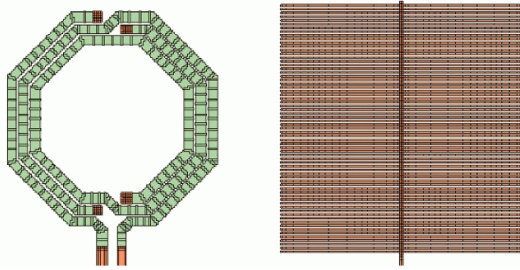


図 31 改良したサブセクション

次に、インダクタの巻線を分割している conformal サブセクションの大きさを制限しましょう。これは導体図形のプロパティの中の”Maximum length”パラメータで設定します。

xmax,ymax ではセルサイズを単位としましたが、このパラメータは長さの単位で設定される事に注意してください。

これらの設定によりサブセクションの様子は図 31 の様になり、よい精度が得られます。

- インダクタの巻線導体とグランドシールド導体が対向している領域では
  - － インダクタの巻線導体に短い conformal サブセクション
  - － グランドシールド導体に小さな staircase サブセクション
- そうでない領域では、グランドシールド導体に大きな staircase サブセクション

## 11 応用例:多数のビアの集合を持つスパイラルインダクタ

インダクタの Q をあげるため、巻線導体を複数の層で構成し、巻線全体に多数のビアホールを配置して複数の層の巻線導体を一体の分厚い導体とすることがあります。このビア集合は、異なる導体層の間に電流を流します (z 方向に) が、インダクタ導体それ自身の断面には影響を与えません。多数のビアの集合それ自身はインダクタの Q や等価抵抗を変化させないのです。

それゆえ巻線導体全体にちりばめられた多数の

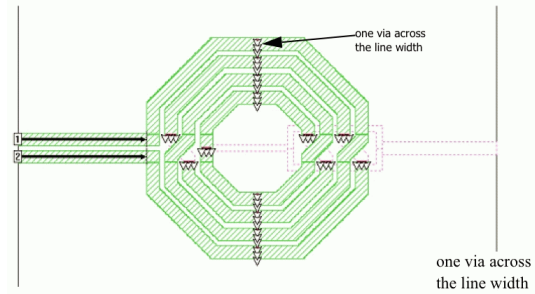


図 32 ヴィアを限定された位置だけに配置した多層導体スパイラルインダクタ

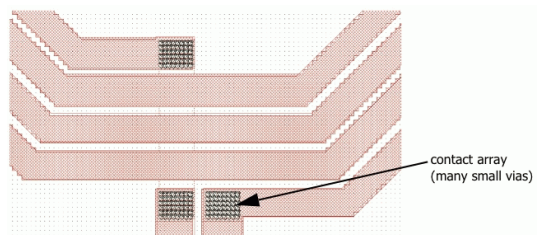


図 33 アレイ状に配置されたビア

ビアの集合は、例えば 90 度毎に (波長より短い間隔で) 配置された少数のビアに置き換えることができます。

ビアを特定の位置だけに配置すればモデルはずっと扱いやすくなります。第一にビアの近傍では conformal サブセクションが使えないのですが、ビアの位置が限定されれば、ビアの無い巻線部分に conformal サブセクションを使えるようになります。第二に、巻線全体にちりばめられた数千のビアは、ソネットの画像表示の速度さえも遅くするので、ビアの数が減れば操作性も向上します。ビアの集合を特定の場所だけに配置した例を図 32 に示します。

図 33 のようなアレイ状に配置されたビアの集合はそのまま解析できなくもありませんが、単独の大きなビアに置き換える事で解析効率を改善します。

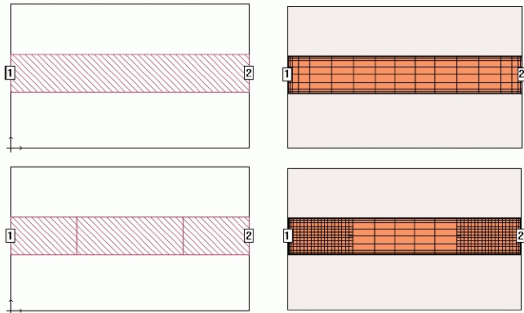


図 34 ダミー導体を使って部分的にサブセクションを制御した線路の例

## 12 サブセクションを制御するためのダミーの図形

ソネットは各導体図形の端部のサブセクションの有無と導体図形内部のサブセクションの密度を制御できます。そこで、回路動作上は一体になった導体図形の一部にダミーの導体図形を配置する事で、導体図形内部のサブセクションを部分的に制御することができます。図 34 の例では、図の上側の二つの図は単純な一本の伝送線路で、下側の二つの図は伝送線路を三つの導体図形に分割し、回路の動作を変えないまま、サブセクションの密度を変更してあります。