


太陽 Grant Thornton Advisory Insights

M&A アドバイザリー

バックナンバー
はこちらから 

テーマ：財務モデリングの基本 3－財務モデルにおける循環参照の回避と解決策

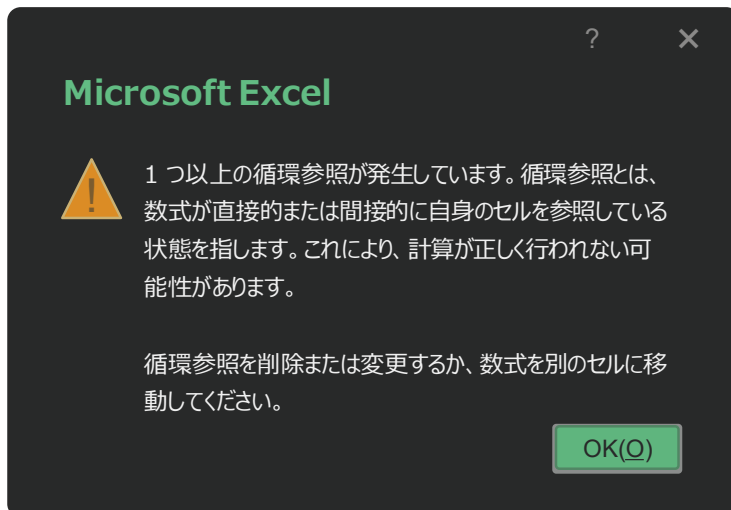
財務モデルを構築する際、誰もが一度は直面するのが「循環参照」の問題です。計算がループしてしまい、Excel にエラーメッセージが表示されると、モデルの信頼性が揺らぎかねません。本稿では、プロフェッショナルな財務モデリングにおいて、循環参照がなぜ起きるのか、そしてどのように解決すべきかを解説します。

循環参照が発生する原因

循環参照は、数式が自分自身の値を参照している場合に発生します。

1. 計算構造の問題

解決しようとしている計算自体が本質的に循環している場合です。答えが問いに依存し、問いがまた答えに依存する構造になっていることがあります。



2. 意図しない計算ループ

モデルの作成過程で、誤って計算の輪をつくってしまうケースです。

具体的には、借入金や現預金の受取利息、デット・サービス・リザーブ・アカウント (DSRA) の計算などでよく発生します。たとえば、期末の現預金残高 (Cash c/f) に基づいて受取利息を計算し、その利息をまた期末残高に加算するようなケースが該当します。以下に簡単な例を示します。

	A	B	C	D
1	総勘定元帳 (Control account)		Year1	
2		期首現金残高 (Cash b/f)	100	
3		現金変動額 (Cash movements)	11	
4		現金預金利息 (Cash interest)	B	-
5		期末現金残高 (Cash c/f)	C	111
6				
7	利息計算 (Interest calculation)			
8		利率 (Rate)	5.0%	
9		預金平均残高 (Average cash balance)	A	106
10		現金預金利息 (Interest)	B	5.3
11				

② 現金預金利息を計算

B = 預金平均残高 (A) × 利率 (5.0%)

③ 期末残高を計算

C = 預金平均残高 (A) + 現金預金利息 (B)

- A: 預金平均残高
= (期首残高 + 期末残高) ÷ 2
- B: 現金預金利息
= 預金平均残高 (A) × 利率
- C: 期末現金残高
= 預金平均残高 (A) + 現金預金利息 (B)



上図の例では、現金預金利息の計算において **循環参照 (依存関係のループ)** が発生する構造を示しています。

まず、期首現金残高と期末現金残高の平均から平均残高 (A) を算出し、これに利率を乗じて現金預金利息額 (B) を計算します。一方、期末現金残高 (C) は、平均残高 (A) に現金預金利息額 (B) を加算して算定されるため、**A → B → C → A** という計算依存のループが生じていることが分かります。

「反復計算」機能による解決とそのリスク

最も簡単な解決策は、Excel のオプションから「反復計算を行う」にチェックを入れることです。

Excel のオプション

全般

数式

データ

文章校正

保存

言語

簡単操作

数式の計算や処理、エラー処理に関するオプションを変更します。

計算方法の設定

ブックの計算 ①

自動(A)

データテーブル以外自動(D)

手動(M)

ブックの保存前に再計算を行う(AV)

反復計算を行う(I)

最大反復回数(X): 100

変化の最大値(C): 0.000

1. リスクの所在

反復計算を有効にするとエラーメッセージが表示されなくなります。そのため、別の場所で誤って循環参照が生じても気づくことができず、エラーの特定が非常に困難になります。

2. 動作への影響

複数ファイル開いている場合など、計算負荷が高まり PC 動作が遅くなるリスクがあります。

推奨される代替アプローチ

プロフェッショナルな現場では、安易に反復計算に頼らず以下の手法を検討します。

1. 計算の簡略化と設計の見直し

受取利息が P&L・Cash Flow 全体に与える影響が軽微であれば、期中平均残高ではなく「期首残高」に利率を掛ける方法に変更します。精度とのトレードオフにはなりますが、循環参照を根本から排除でき、意思決定に支障がない場合は有効な戦略です。

2. 代数的な解決

数式を組み替えることで、数学的に循環を解消します。

① 基本式の作成：利息 = (期首残高 + 期末残高) / 2 × 利率

② 代入と整理：利息 = (2 × 期首残高 + キャッシュ変動額) / (2 / 利率 - 1)

ただし、この方法はモデル利用者にとって直感的ではなく、複雑なモデルでは適用が難しい場合があります。

3. VBA（マクロ）の活用

私たちが推奨するのは、マクロを利用する方法です。

1) マクロの仕組み

計算結果を一度コピーし、別のセルに値として貼り付ける処理を繰り返します。貼り付けた数値と計算結果が一致した時点で処理を停止させます。

2) 活用のメリット

モデルの計算負荷を抑えられるだけでなく、「実行ボタン」を設置することで、新しいユーザーに対しても「ここで循環計算が行われている」ことを明確に示せます。

下図のモデルでは、計算上の循環参照を回避するため、数式による自動計算の中に「マクロによる値の固定」という工程を挟み込み、計算を完結させています。

	A	B	C	D	E	F	G	H	I	J	K	L	M	
1	Circularity problem solved													
2														
3	Control account								Year 1	Year 2	Year 3	Year 4		
4														
5	Cash b/f		JPY 000						100	116	158	213		
6	Cash movements		JPY 000						11	35	45	23		
7	Cash interest		JPY 000					A	5.4	6.9	9.3	11.5	<<参照元は、「エリアD」	
8	Cash c/f		JPY 000						116	158	213	247		
9														
10	Interest calculation													
11														
12	Rate		%						5.0%					
13	Average cash		JPY 000						108	137	185	230		
14	Interest		JPY 000					B	5.4	6.9	9.3	11.5		
15														
16	Macro copy area		JPY 000					C	5.4	6.9	9.3	11.5	vbCopy	
17	Macro paste area		JPY 000					D	5.4	6.9	9.3	11.5	vbPaste	
18	Absolute variance		JPY 000					E	0.0	0.0	0.0	0.0		
19														
20														
21	Tolerance		JPY 000						0.0001	interest circular				
22	(the macro will stop once the variance is below this tolerance)													
23														
24	Flag to end macro	check							OK					
25													vbVariance	

(1) 計算のステップ

まず、モデル内の各エリアがどのように繋がっているかを整理します。

【起点】 エリア A：利息の計上（リンク）

エリア D（値として貼り付けられた利息）を参照し、現預金の期末残高に反映させます。

【中間】 エリア B：平均残高に基づく利息の計算

「期首残高」と、エリア A の影響を受けた期末残高の平均残高に基づき、利息額を計算します。

【計算】 エリア C：理論利息の算出

エリア B で計算された利息残高を参照します。

(2) マクロによる「循環の切断」と「収束」

通常、「エリア C」の結果をそのまま「エリア A」に戻すと循環参照エラーになります。そこで、マクロが以下の「橋渡し」を担います。

① コピー＆ペースト（値貼り付け）

マクロが「エリア C」をコピーし、「エリア D（定数入力欄）」に値として貼り付けます。

② 循環の遮断

「エリア D」は「数式」ではなく「数値」であるため、「エリア D」から「エリア A」へデータが反映される際、数式の繋がりが一度切断されます。これにより、Excel 上での循環参照エラーを回避できます。

③ 収束（イテレーション）

貼り付けによって「エリア A」の数値が変わると、「エリア B・C」も再計算されます。マクロは「エリア C」と「エリア D」の数値が完全に一致するまでこの工程を繰り返し（収束）、最終的にすべての整合性が取れた時点で停止します。

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Circular reference example		Circular reference error check										
2									Out-turn	Forecast	Forecast	Forecast	
3								Period	FY22	FY23	FY24	FY25	
4								Start date	1-Jan-22	1-Jan-23	1-Jan-24	1-Jan-25	
5								End date	31-Dec-22	31-Dec-23	31-Dec-24	31-Dec-25	
6	Using Macro												
7	Control account												
8													
9			Cash b/f	JPY 000					100	116	158	213	
10			Cash movements	JPY 000					11	35	45	23	
11			Cash interest	JPY 000					5.4	6.9	9.3	11.5	
12			Cash c/f	JPY 000					116	158	213	247	
13													
14	Interest calculation												
15													
16			Rate	%					3.0%				
17			Average cash	JPY 000					108	137	185	230	
18			Interest	JPY 000					3.2	4.1	5.6	6.9	
19													
20			Macro copy area	JPY 000					3.2	4.1	5.6	6.9	vbCopy
21			Macro paste area	JPY 000					5.4	6.9	9.3	11.5	vbPaste
22			Absolute variance	JPY 000					2.2	2.7	3.7	4.6	
23													
24													
25			Tolerance	JPY 000					0.0001				
26			(the macro will stop once the variance is below this tolerance)										
27													
28			Flag to end macro	check					ERROR				
29									vbVariance				

計算の前提条件（例：利率）・その他の数値が変更された場合には、マクロを再実行する必要があります。エラーメッセージ（Circular reference error check 「循環参照エラーチェック」）を設定しておくことで、ユーザーに対してマクロの再実行を促し、数値を正確に保つことが可能となります。

以下でご確認いただける通り、コード自体は非常にシンプルです（なお、マクロのコード内には、ロジックを説明する注釈を常に付すべきです）。また、入力値を変更するたびにマクロを実行しなければならないことをユーザーに注意喚起するため、視認性の高い実行ボタンを設置しています。

```
(General) | interestcircular
Option Explicit
Sub interestcircular() 'Macro to resolve circular references
    Do Until Range("vbVariance") = 0 'Repeat the calculation until the variance between the paste values and copy values is zero
        Range("vbPaste").Value = Range("vbCopy").Value
        Application.Calculate
    Loop
End Sub
```

AI の利用で、コード記述に慣れていない人でも瞬時にコードを記述することが可能です。

（参考）Google Gemini による簡易的なコード（実際の計算構造、前提条件により微修正は必要となる点には留意されたい）

1. 最小構成の循環参照解消コード

```
</> VBA
Sub SolveInterestSimple()
    ' I29:L29(計算式)とI30:L30(値)に差がある間、値を転記し続ける
    Do While WorksheetFunction.SumProduct(Abs(Range("I29:L29").Value - Range("I30:L30").Value) > 0)
        Range("I30:L30").Value = Range("I29:L29").Value
    Loop
    MsgBox "再計算完了"
End Sub
```

この手法の大きな利点は、Excel の「反復計算」機能を有効にする代替案と比較して、モデルへの継続的な計算負荷を排除できる点にあります。つまり、マクロのボタンが押された時にのみ計算が行われるため、モデルの動作を軽く保つことができます。

お見逃しなく！

財務モデルを扱う以上、循環参照との付き合い方は避けて通れません。設計の見直しやマクロの活用を検討し、単にエラーを消すだけでなく、透明性が高くメンテナンスしやすいモデルを目指すべきです。

計算前提（利率等）を変更した際は必ずマクロを再実行し、エラーチェックセルが OK になっているか確認することを忘れないでください。