

BGSP Ver. 2.2

ユーザーズ・マニュアル

BGSP 研究会

田中和博(九州工業大学名誉教授)

肥後 寛(九州工業大学)

山本耕治(九州工業大学元客員教授)

中田 肇(東京電機大学名誉教授)

目次

1. はじめに	3
1-1 BGSP2.2のセットアップ	
1-2 BGSP2.2の起動	
1-3 コンパイラの設定	
1-4 BGSP2.2のアンインストール	
2. ボンドグラフ	4
2-1 基本変数	
2-2 基本素子	
2-3 ボンドグラフによる配管表現例	
3. ボンドグラフの作成	7
3-1 メイン画面	
3-2 素子の配置	
3-3 ボンドの接続	
3-4 ボンド, 素子の削除	
3-5 メニューバーの概要	
4. 各種入力	11
4-1 特性式及びパラメータの入力	
4-2 計算条件の入力	
5. 解析	12
5-1 計算方法	
5-2 計算結果の表示	
5-3 BGS ファイルの実行	
6. 実験値入力の方法	15
6-1 BGE ファイルを開く	
6-2 SE1 素子に実験値入力を設定する	
6-3 計算を行う	
6-4 計算結果	
6-5 実験値入力ファイルのフォーマット	
7. その他の機能	22
7-1 変数チェック機能	
7-2 グローバル変数 XX(i)	
7-2-1 再帰型ループの解消	

7-3 temp.c を利用したデバッグ作業

8. 状態方程式の導出機能	27
8-1 状態方程式の導出とチェック	
9. BGSP2.0以降における制限	29
9-1 素子数, ボンド数, パラメータ数の制限	
9-2 使用禁止変数	
9-3 その他の制限	
10. 例 1 マス・ばね・ダンパーにおけるボンドグラフ	30
10-1 C version の場合	
10-2 Fortran Version の場合	
10-3 計算	
11. 例 2 油圧システムにおけるボンドグラフ	39
11-1 C version の場合	
11-2 Fortran Version の場合	
11-3 計算	
12. 例 3 電気回路のボンドグラフ	50
12-1 低域通過フィルター	
12-1-1 C version の場合	
12-1-2 Fortran version の場合	
12-1-3 計算	
12-2 高域通過フィルター	
12-2-1 C version の場合	
12-2-2 Fortran version の場合	
12-2-3 計算	
13. BGSP2.2 のご利用にあたって	62
付録. 1 Fortran と C++の比較	63
付録. 2 ボンドグラフ(BG)に関する基礎事項	67
付録. 3 ボンドグラフシミュレーションツールの考え方	70
付録. 4 BGSPVer.2.0 以降のファイル構造について	82

BGSP 改訂履歴

2023 年 8 月 BGSP_Version2.0 公開

ユーザーインターフェイスの更新

2024 年 6 月 BGSP Version2.1 公開

デバッグ機能の更新

- ・計算実行時, CBGSP の一時停止による BGS ファイルの不備の確認

- ・計算実行時, コンパイルエラーのファイル出力

2024 年 10 月 BGSP Version2.2 公開

- ・計算用中間ファイル(temp.c, temp.BGS)の生成

- ・特性式の最後に改行がある場合に実行できなくなるバグの修正

- ・その他の軽微なバグの修正

1. はじめに

1-1 BGSP2.2 のセットアップ

事前に C 又は Fortran の任意のコンパイラをインストールする。

推薦コンパイラ: ボーランド C

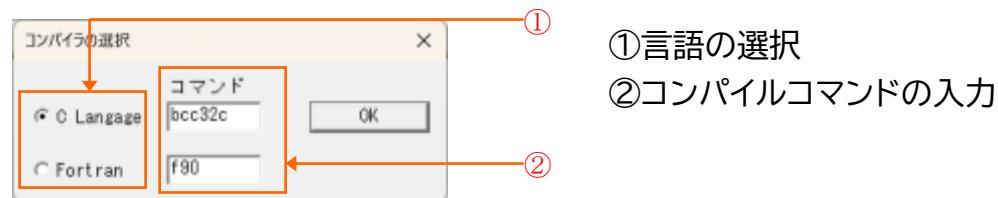
BGSP2.0 のシステムが入ったフォルダをインストールしたいディレクトリにペーストする。

1-2 BGSP2.2 の起動



「スタート」ボタン - 「プログラム」 - 「BGSP」 - 「BGSP」をクリックすると BGSP が起動します。

1-3 コンパイラの設定



本ソフトウェアは C 言語と Fortran に対応しています。

①から使用する言語を選択し、②に使用するコンパイルコマンドを入力します。コンパイルコマンドは使用する環境に適したコマンドを入力して下さい(以降、C 言語の場合を C version, Fortran 言語の場合を Fortran version とします)。

注)デフォルトで BorlandC のコンパイルコマンド“bcc32c”と Visual Fortran のコンパイルコマンド“f 90”が入力されています。

1-4 BGSP2.2 のアンインストール

1-1でインストールしたフォルダを削除してください

2. ボンドグラフ

2-1 基本変数

ボンドグラフの基本変数と物理量との関係を表1に示します。

表1 ボンドグラフの基本変数と物理量

基本変数(記号)	電気系	流体系	直線運動系	回転運動系
エフォート(e)	電圧	圧力	力	トルク
フロー(f)	電流	流量	速度	角速度
モーメンタム(MN)	磁束	圧力モーメント	運動量	角運動量
ディスプレースメント(DP)	電荷	体積	変位	角変位

モーメンタム : エフォートの時間積分

ディスプレースメント : フローの時間積分

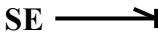
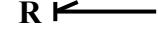
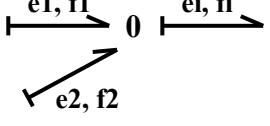
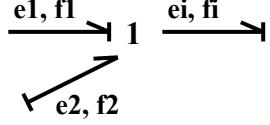
2-2 基本素子

ボンドグラフを構成する基本素子および特性関数を表2に示します。

注意事項

- ①代入式の左辺変数は、素子ごとに決まっています。(例 SE:E= , C:C=)
- ②代入式の右辺において、Tは時間(sec)変数です。
- ③代入式の右辺において、zは素子によっていろいろな変数となります。(次頁参照)

表2-1 ボンドグラフを構成する基本素子および特性関数

素子	記号	特性関数定義式の形式		
		代入文	左辺変数の意味	変数 z の意味
SE 素子		$E = e$	エフォート	
SF 素子		$F = f$	フロー	
R 素子		$R = e(z)$	エフォート	フロー
		$R = f(z)$	フロー	エフォート
C 素子		$C = e(z)$	エフォート	ディスプレーイスメント
I 素子		$L = f(z)$	フロー	モーメンタム
TF 素子		$TF = 1/n$	変換比 $e2 = e1/n, f1 = f2/n$ $e1 = n \cdot e2, f2 = n \cdot f1$	
		$TF = n$		
GY 素子		$GY = 1/m$	変換比 $f2 = e1/m, f1 = e2/m$ $e1 = m \cdot e2, e2 = m \cdot f1$	
		$GY = m$		
0 接点		$f1 + f2 \dots -fi = 0$ $e1 = e2 \dots = ei$	i 個あるボンドの中で、0側にストロークが 1本だけ付く。	
1 接点		$f1 = f2 \dots = fi$ $e1 + e2 \dots - ei = 0$	i 個あるボンドの中で、1側にストロークが 1本だけ付かない。	

SE 素子, SF 素子, C 素子, I 素子はボンドが 1 つだけ付けられます。

TF 素子, GY 素子はボンドが2つまで付けられます。

0 接点, 1 接点はボンドが8つまで付けられます。

2-3 ボンドグラフによる配管表現例(付録2参照)

図2-1の油圧回路はボンドグラフで表すと図2-2のように表現されます。

- ①. SF はポンプであり, 供給流量 Q_1 を意味します.
- ②. C は配管の容積効果です.
 C の特性式内での変数 z は, $\int (Q_1 - Q_2) dt$ です.
- ③. I は配管の流体の慣性です.
 I の特正式内での変数 z は, $\int (P_2 - P_3 - P_r) dt$ です.
- ④. R は配管の抵抗です.
 R の特正式内での変数 z は, Q_2 です.
- ⑤. SE はタンクであり, タンク圧 P_3 を意味します.
- ⑥. 0 接点で圧力と流量の関係は, $P_1 = P_2 = P_c, Q_c = Q_1 - Q_2$ となります.
- ⑦. 1 接点で圧力と流量の関係は, $P_i = P_2 - P_3 - P_r, Q_2 = Q_3 = Q_c = Q_r$ となります.

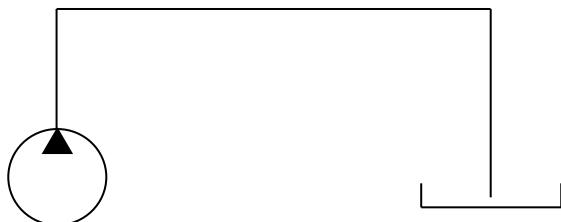


図2-1 油圧回路

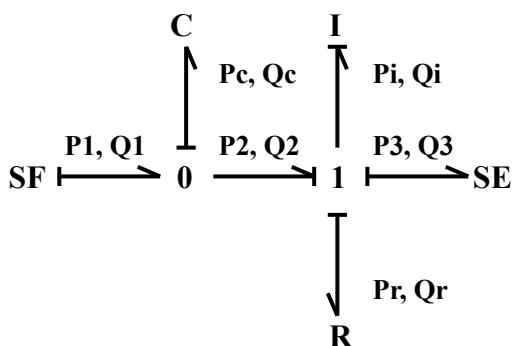
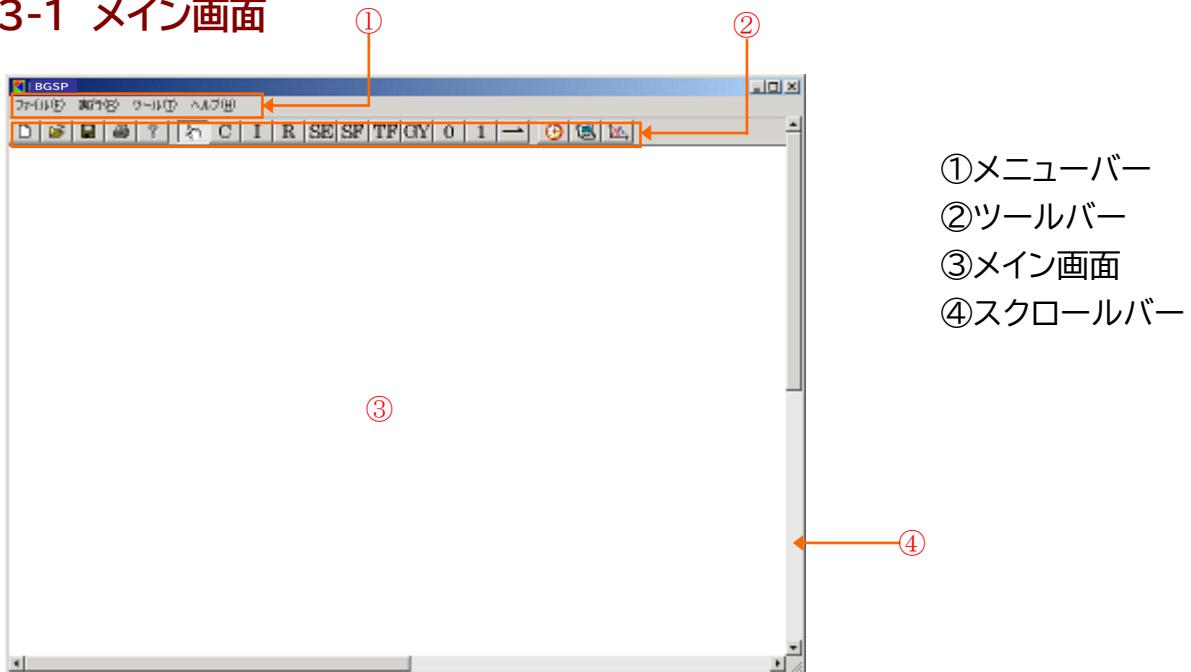


図2-2 配管のボンドグラフ表現

3. ボンドグラフの作成

3-1 メイン画面



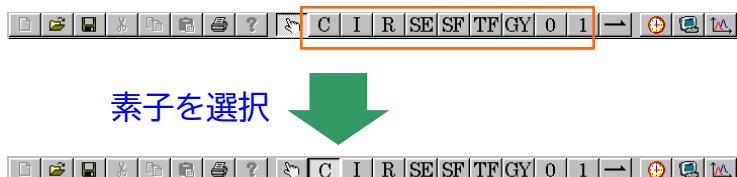
①メニューバー

②ツールバー

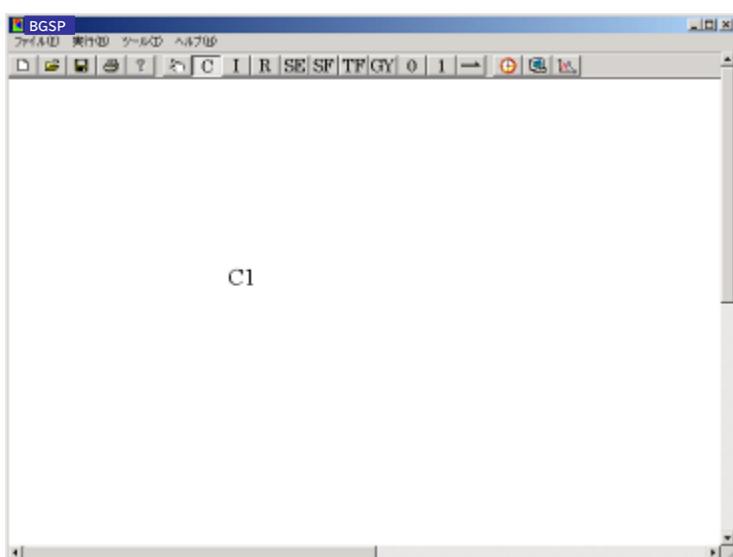
③メイン画面

④スクロールバー

3-2 素子の配置



ツールバーから目的の素子を左クリックすると、選択したツールバー上の素子がくぼみます。

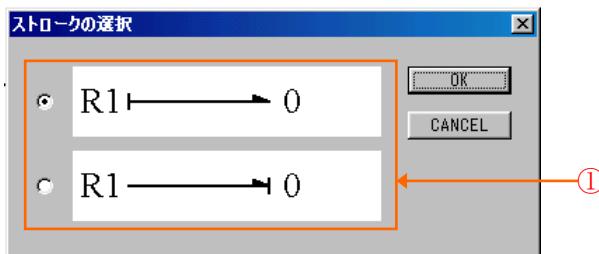


この状態でメイン画面の任意の位置を左クリックすると、クリックした場所に素子が配置されます。そのとき、素子に番号が自動的に付きます。

3-3 ボンドの接続



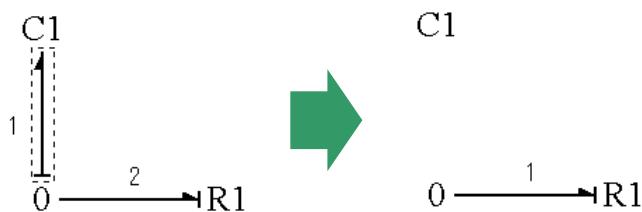
素子を接続するには、まずツールバーからボンドを左クリックします。次に、メイン画面上の接続元となる素子を左クリックし、その後、接続先の素子を左クリックするとボンドが接続されます。そのとき、自動的にボンドに番号が付きます。



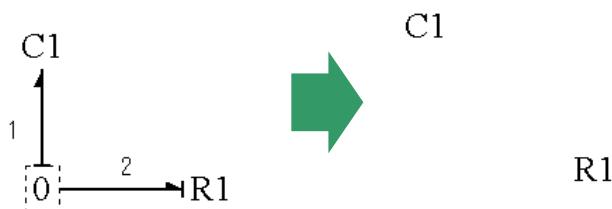
①ストロークの選択

ボンドを接続する際、C, I, SE, SF 素子は自動的にストロークが決定されますが、R, TF, GY 素子と 0, 1 接点は自動的にストロークを決定することが出来ないので、この場合はユーザがストロークの位置を決定して下さい。

3-4 ボンド、素子の削除



ボンドを削除するには、まずツールバーから手のアイコンを左クリックします。次に、メイン画面上の削除するボンドを選択してから、キーボード上の“Delete”キーを押すとボンドが削除されます。

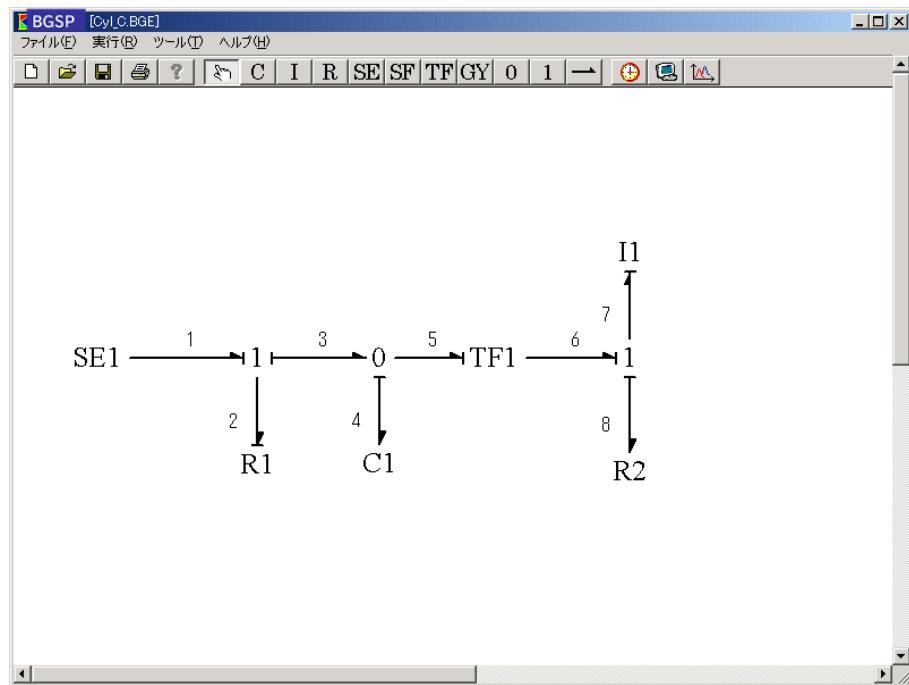


また、素子も同様の方法で削除することができます。素子を削除するときは、素子に接続しているボンドも同時に削除されます。

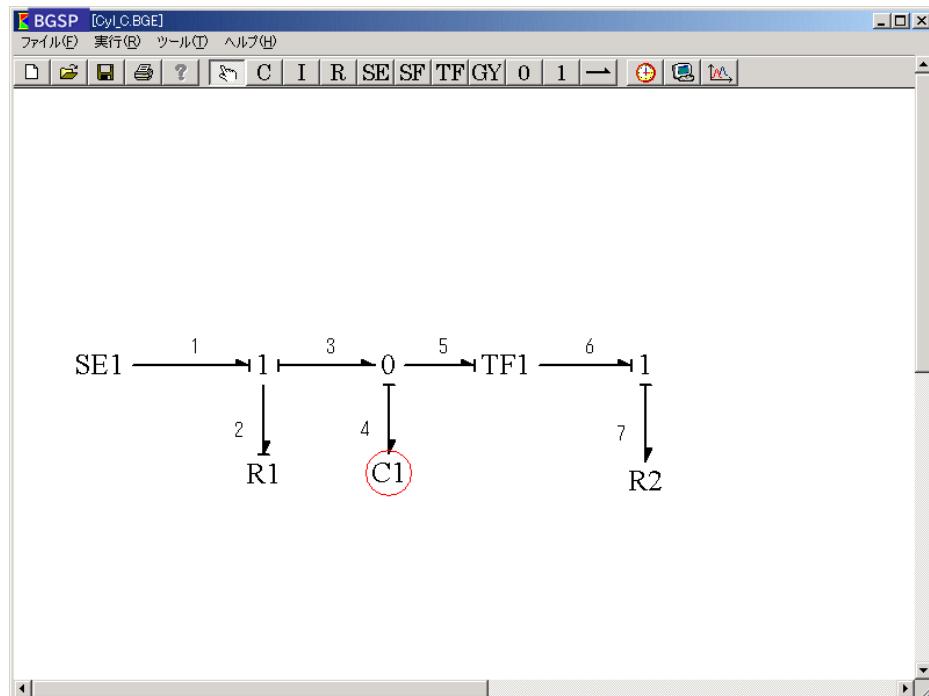
素子、ボンドを削除すると「出力変数の決定」ダイアログボックスの値が自動的に削除されます。

また、素子及び、ボンドを削除したときに修正すべきフィードバック変数が定義されている素子に赤い丸を表示する機能があります。以下にその例を示します。

下記のシステム・ボンドグラフ(システムのボンドグラフ・モデル)では、ボンド番号 7 番の I1 素子の Displacement が C1 素子のフィードバック変数に設定されています。



この状態で I1 素子を削除するとボンド番号 7 番も消えるため、フィードバック変数が定義されている C1 素子に赤い丸(Warning)が表示されます。



この赤い丸(Warning)は C1 素子のフィードバック変数を修正する必要があることを示します。この赤い丸は、素子を移動したり、特性式入力画面を開くと消えます。

3-5 メニューバーの概要

ここでは、メニューの機能について説明します。メニューバーには、[ファイル]、[実行]、[ツール]、[ヘルプ]の4つのメニューがあります。それぞれの機能を以下に示します。

[ファイル]

新規作成…ボンドグラフを新規作成します。
ファイルを開く…BGE ファイルを開きます。
名前を付けて保存…編集中のボンドグラフに名前をつけて保存します。
上書き保存…編集中のボンドグラフを上書き保存します。
画面を保存…編集中の画面を保存します。
終了…プログラムを終了します。

[実行]

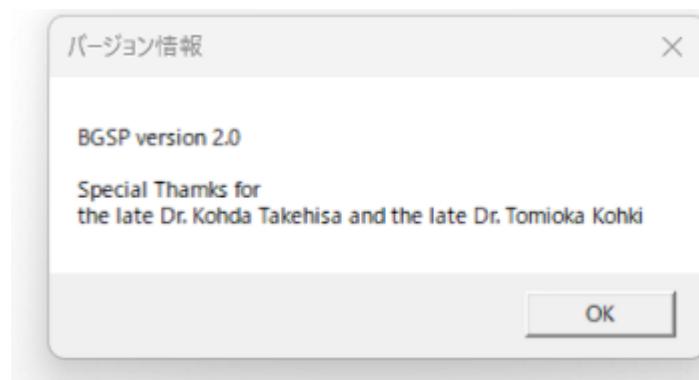
ボンドグラフの実行…現在編集中のボンドグラフを実行します。
BGS ファイルから実行…BGS ファイルを実行します。

[ツール]

実験値入力…実験値入力の設定をします。
コンパイラの選択…コンパイラを選択します。
変数のチェック…特性式入力画面で設定した変数をチェックします。
方程式を表示…現在編集中のボンドグラフの状態方程式
(連立微分方程式、または連立積分方程式のどちらかを選択する)

[ヘルプ]

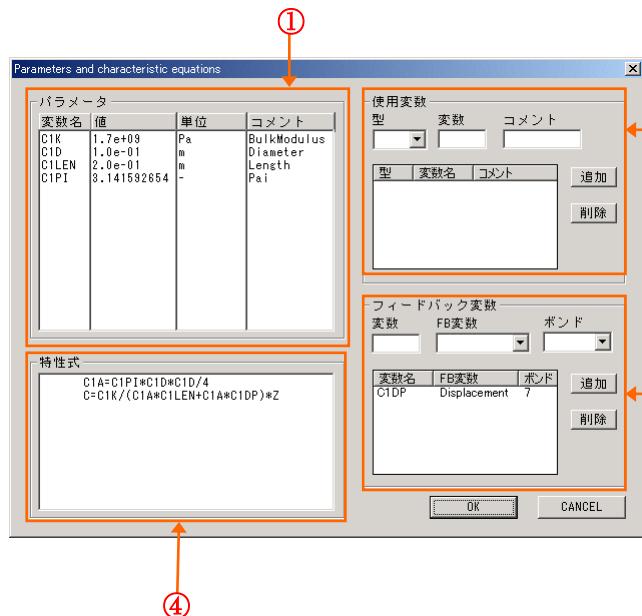
ヘルプの表示…ヘルプを表示します。(今後、充実する予定です。)
バージョン情報…現在実行しているプログラムのバージョンを表示します。



BGSP(Ver. 1)の作成に貢献した故 幸田武久博士と
BGSP(Ver. 2)の作成に貢献した故 富岡弘毅博士に
敬意を表して実名を表記しています。

4. 各種入力

4-1 特性式及びパラメータの入力



① パラメータ入力部

変数名:5 文字以下で定義

値:標記の仕方は自由

単位:コメント行に挿入される

コメント:コメント行として表示

② 使用変数入力部

(C 言語の時のみ入力)

型を選択, 変数を入力(コメントは自由)

した後、「追加」ボタンを押すと一覧表示部(②の下部)に登録される。「削除」ボタンで削除できる。

③ フィードバック変数入力部

上記同様.

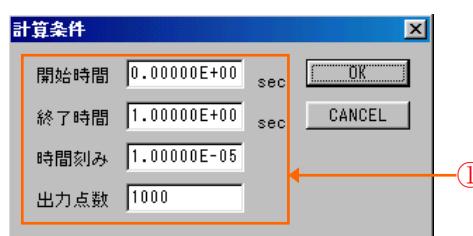
④ 特性式入力部

C 言語, Fortran 言語の決まりに従って入力する。

メイン画面に配置された素子を左クリックして特定する。その後、右クリックすると特性式入力画面が出現します。この画面内に選択した素子の特性式やパラメータ、使用変数、フィードバック変数などを入力して下さい。また、特性式は最初に選択した言語の書式に沿った記述をして下さい。

なお、Fortran の場合、特性式は7カラム目からとなりますので注意して下さい。

4-2 計算条件の入力

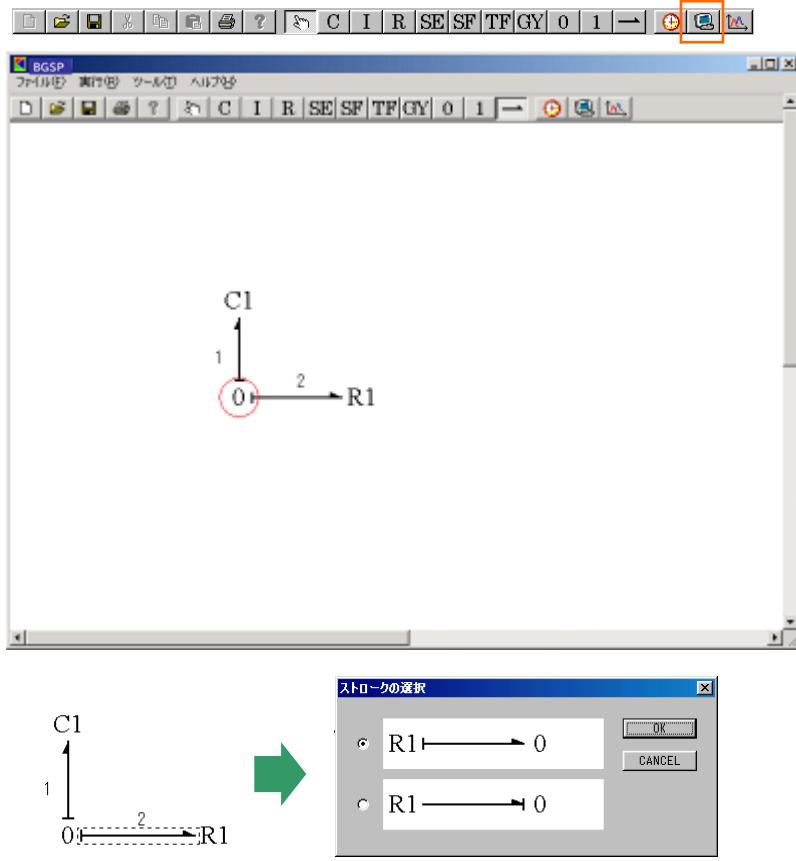


① 計算条件入力部

ツールバーから計算条件のアイコンを左クリックすると、計算条件入力画面が出現します。ここで、計算条件を入力します。(計算)開始時間は常に「0.0sec」にしてください。ここで、出力点数は、(終了時間 - 開始時間) / 時間刻み の値より大きく設定できませんので注意してください。

5. 解析

5-1 計算方法



計算するには、ツールバーからPCのアイコンをクリックします。

もし、ここでボンドグラフに間違いがあれば、赤い丸で表示されるのでストロークを修正して下さい。



ストロークを修正するには、修正するボンドを左クリックして該当するボンドを特定した後、右クリックするとストロークの選択画面が出るので、修正して下さい。

①出力変数設定部

②出力変数表示部

出力変数として、下記の5変数量をセット可能です。

- EFFORT
- FLOW
- MOMENTUM
- DISPLACEMENT
- POWER

ボンドグラフに間違いが無ければ、次に出力変数を決定します。①の出力変数設定部で“出力変数”，“ボンド(番号)”を選択し，“追加”を左クリックします。追加すると②の出力変数表示部に出力変数が表示されます。また、出力変数を削除するには、②から、削除する出力変数を左クリックし、「削除」ボタンを選択します。

そして、“計算開始”を選択すると、計算が開始されます。



しばらくすると, 出力するファイル名を入力する画面が現れるので, ファイル名を入力します(拡張子は必要ありません). ソフトをインストールしたフォルダ内の csv ディレクトリに自動的に.csv ファイルを作成します.

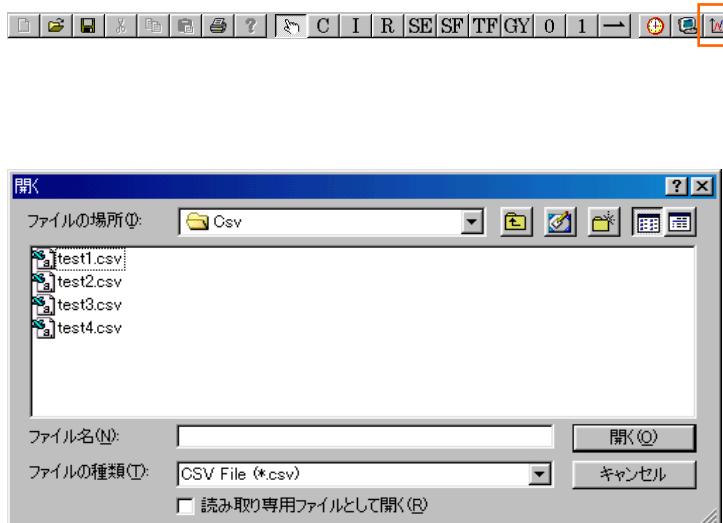


計算が終了すると, メッセージが現れるので, “OK”を左クリックして終了して下さい.

※Version 2.1 以降は計算実行時に設定などの不備をチェックできます.

計算開始ボタンを押したあと, コマンドエディタが立ち上がり, 中間ファイル(cファイル)生成時の過程を確認することができます. コマンドエディタに“Press enter to continue”と表示されたら, リターンキーを押すと“出力ファイル名の入力”の画面が表示され, 計算を開始します. コンパイルの情報は advance→Comp フォルダに格納されます.

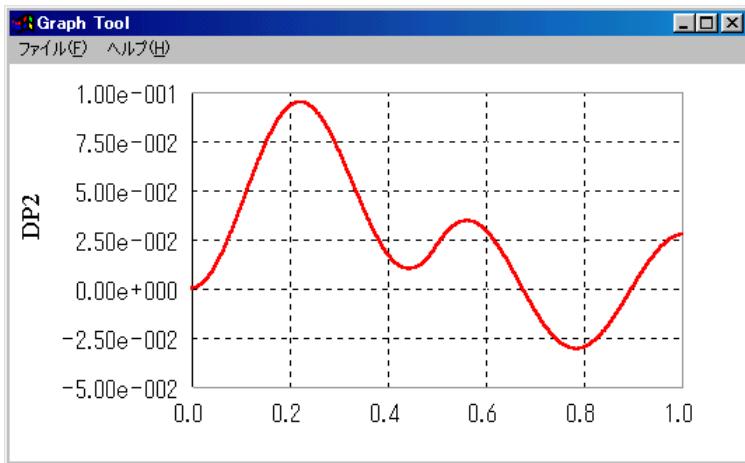
5-2 計算結果の表示



計算結果をグラフに表示するには, ツールバーからグラフのアイコンを選択します. ここで, 先程入力した.csv ファイルを選択し, “開く”をクリックするとグラフが表示されます.

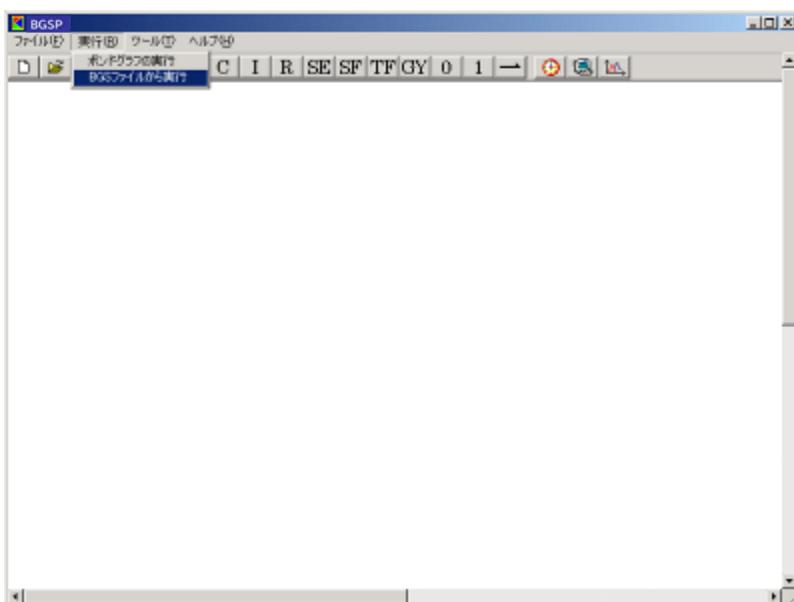
残念ながら, このグラフィック機能は簡易的に作図する機能です.

計算結果の作図に操作を加えるべく高い作図機能が欲しい場合には, CSV 形式でデータが保存されるため, Excel を利用して作図することも可能です.

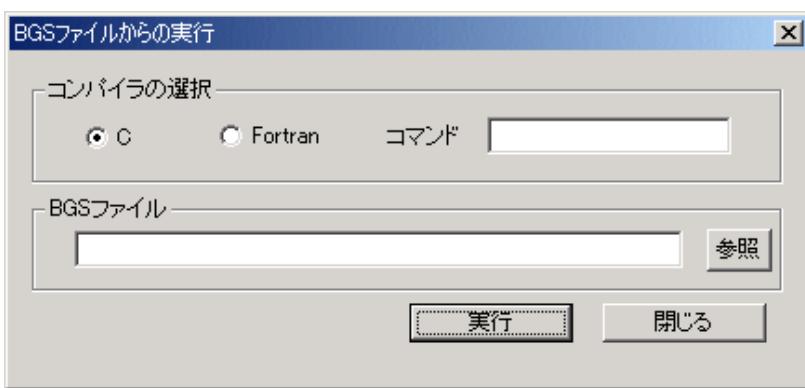


5-3 BGS ファイルの実行

メニューバーの[実行]から BGS ファイルから実行を選択します。



以下のダイアログボックスが表示されます



コンパイラの選択のところで, C 言語で実行する場合は C を Fortran 言語で実行する場合は Fortran を選択し, コマンドをキーボードで入力します. 以下の例ではコンパイラに C 言語, コマンドに cl を指定します. その後, 実行する BGS ファイルの場所を指定します.



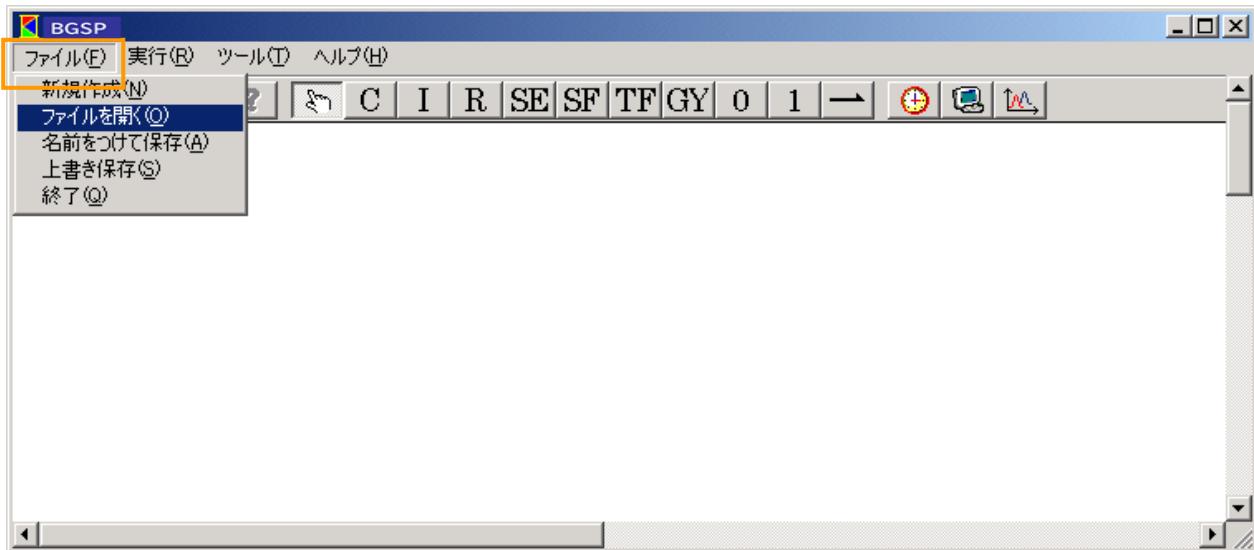
すべて入力したら、「実行」ボタンを押し BGS ファイルを実行します。

6. 実験値入力

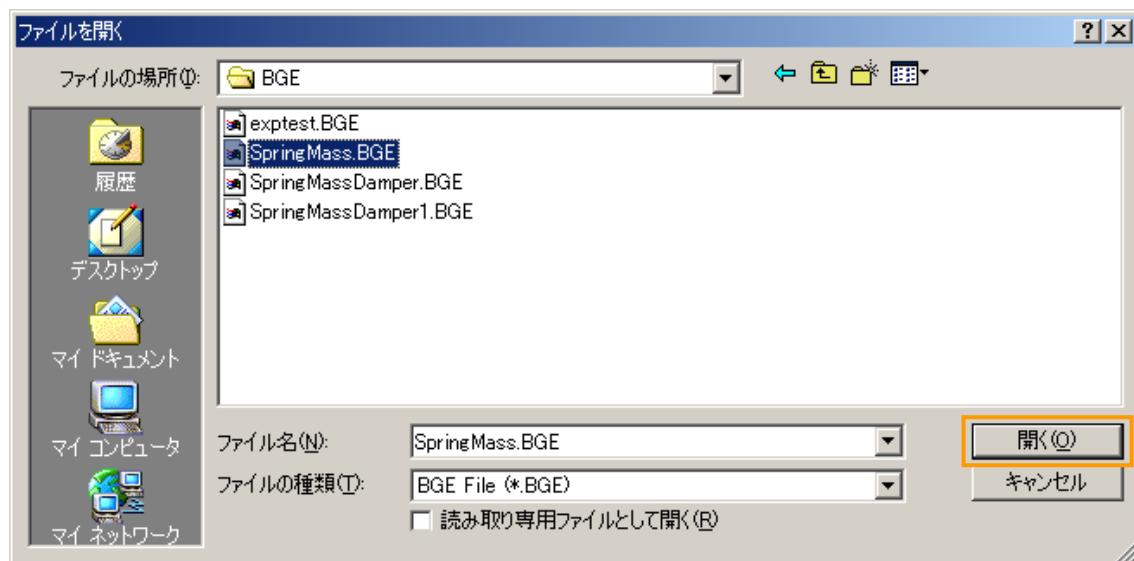
ここでは, C version のばねマス系について, SE 素子に実験値入力する方法を説明します.

6-1 BGE ファイルを開く

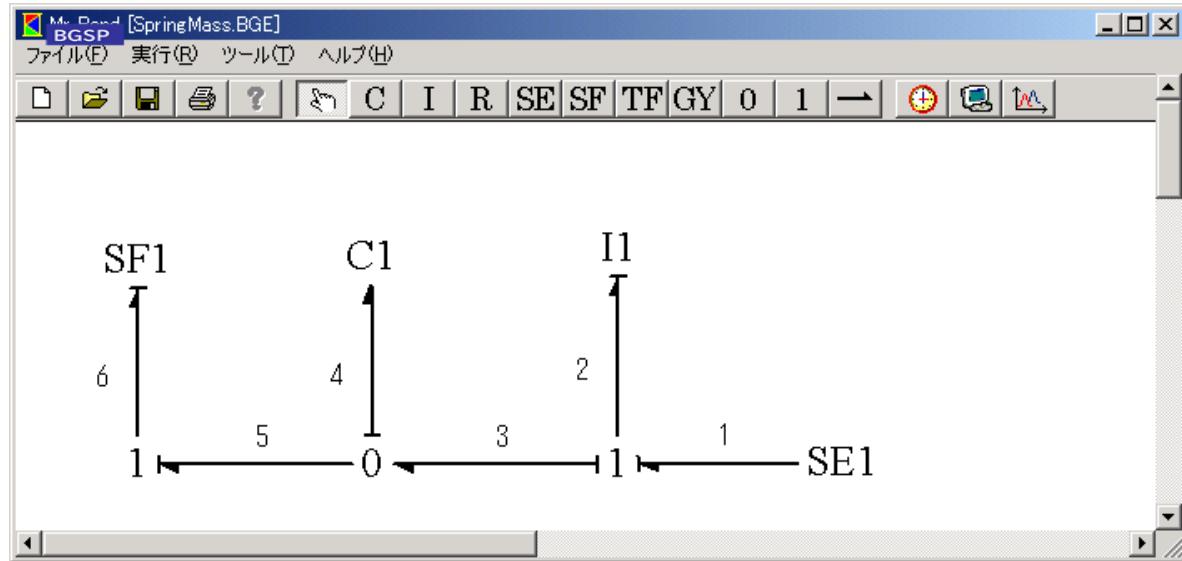
ファイルからファイルを開くを選択します.



以下のダイアログボックスが表示されるので, そこで実験値入力したい BGE ファイル(ここでは, SpringMass.BGE)を選択し, 「開く」ボタンを押します.



「開く」ボタンを押すと以下のボンドグラフがウインドウに表示されます。



このボンドグラフは、ばねマス系の物理モデル(下図参照)と等価です。

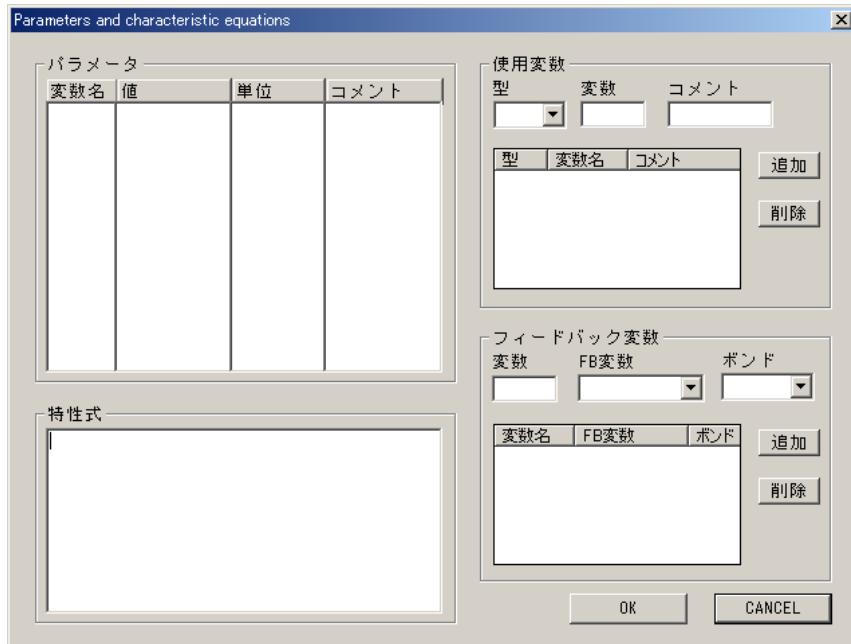


各素子の特性式は以下のようになっています。

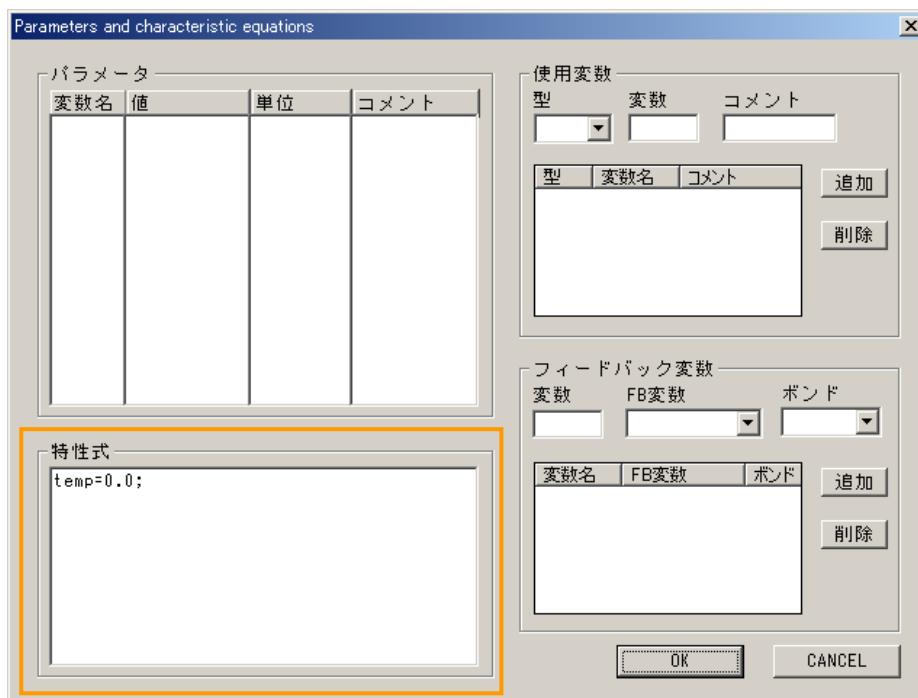
素子名	特性式	使用変数	変数の値
SE1	$E=50;$	なし	50
I1	$L=Z/I1MA;$	I1MA	100 [kg]
C1	$C=C1SP*Z;$	C1SP	63165[N/m]
SF1	$F=0.0;$	なし	0

6-2 SE1 素子に実験値入力を設定する

SE1 を選択します。右クリックをし、特性式ダイアログボックスを開きます。

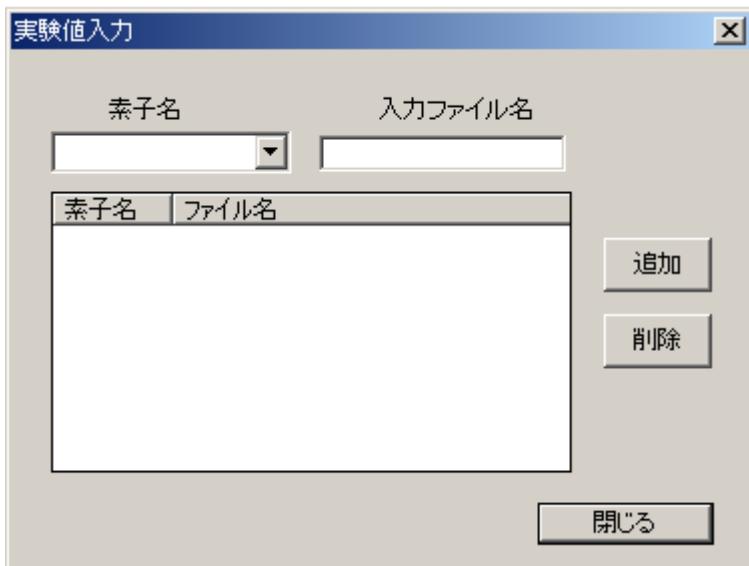
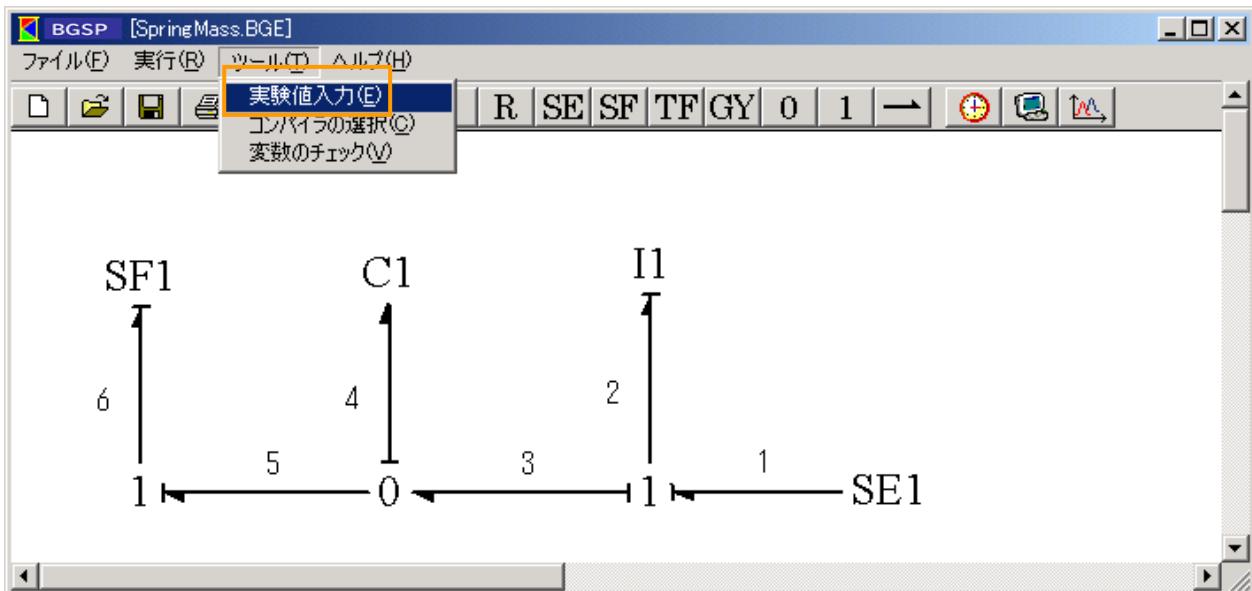


特性式に $temp=0.0;$ と書きます (Fortran version の場合は、 $TEMP=0.0$ となります)。
注: 実験値を設定する素子には、必ず上記の式を特性式に入力してください。

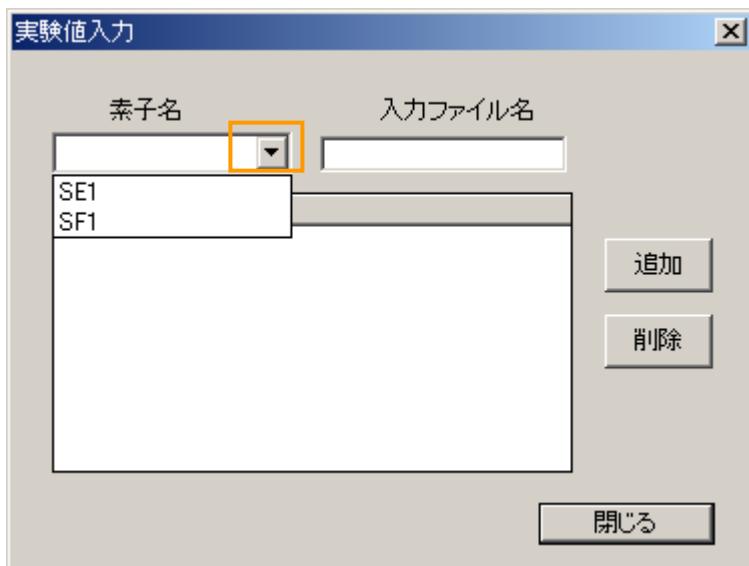


「OK」ボタンを押し特性式ダイアログボックスを閉じます。

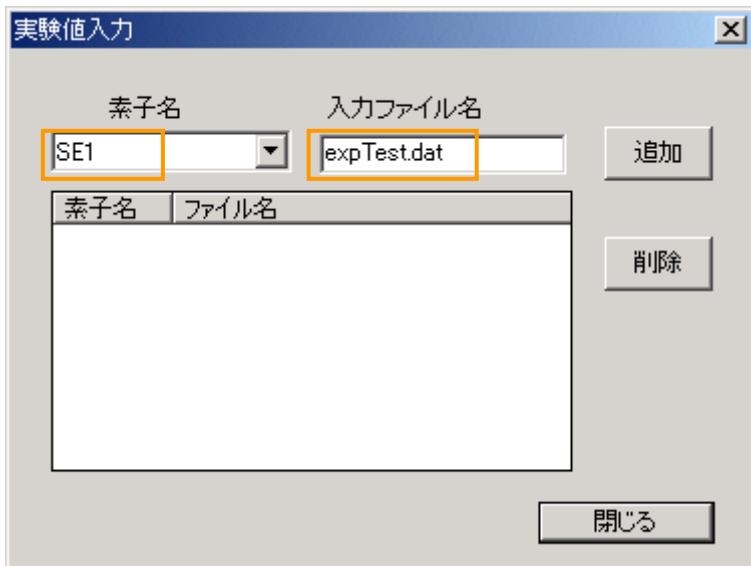
メニューバーのツールから実験値入力を選択します。



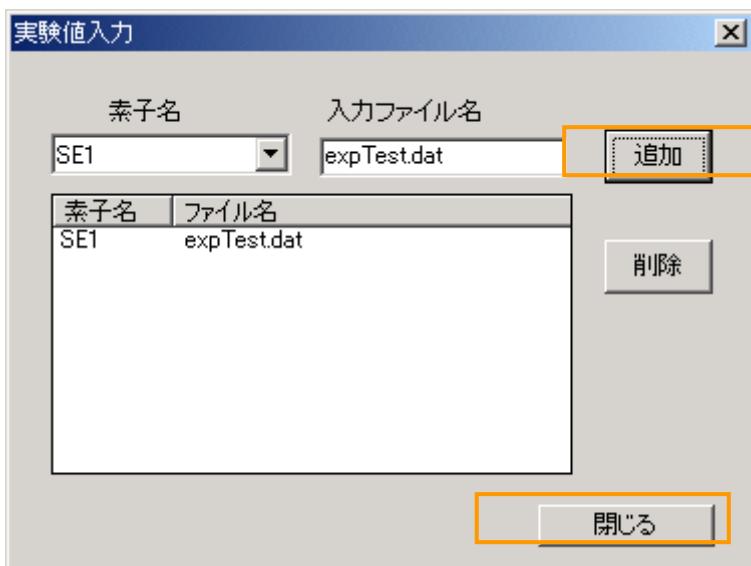
実験値入力ダイアログボックスが表示されます。



素子名の矢印を押すと SE 素子と SF 素子が表示されます。



素子名で SE1 を選択し, 入力ファイル名には expTest.dat とキーボードで入力します.

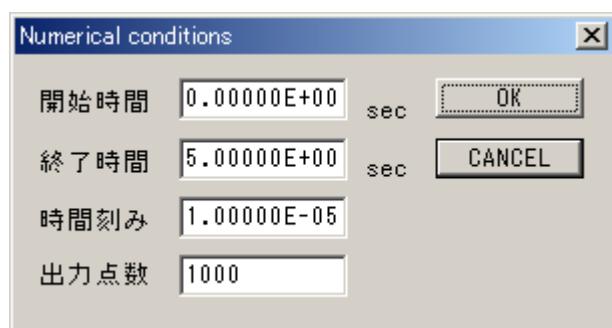


「追加」ボタンを押すと, リストボックスに素子名とファイル名が表示されます.

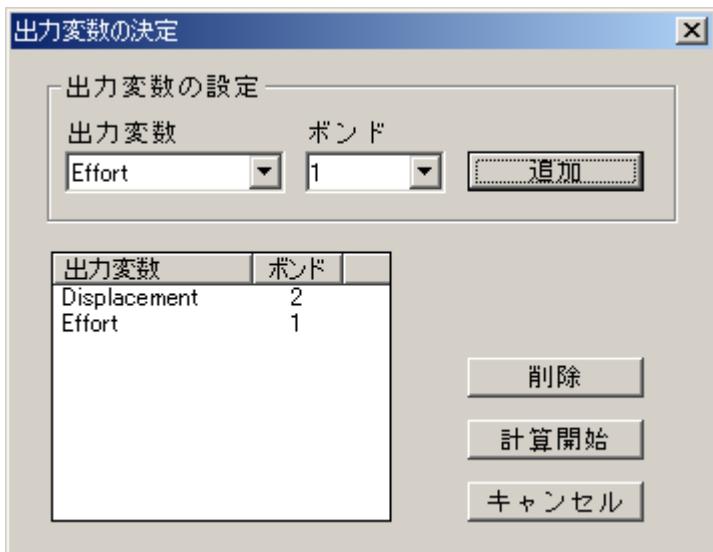
「閉じる」ボタンを押し, 実験値入力ダイアログボックスを閉じます.
これで, 実験値入力の設定が完了します.

6-3 計算を行う

設定時間は以下のようにします.



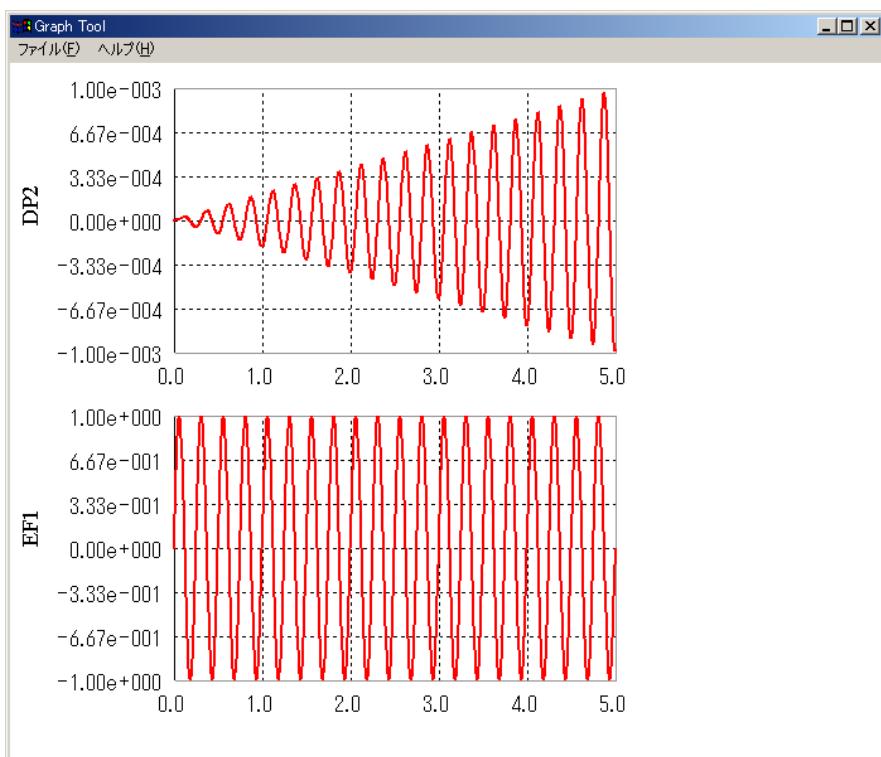
また、出力変数の設定は以下のようにします。



後は、「計算開始」ボタンを押すと計算ができます。

6-4 計算結果

計算結果は以下のようになります。



6-5 実験値入力ファイルのフォーマット

実験値入力による計算をする場合, 実験値入力ファイルをインストールされたフォルダ内にあらかじめ作成する必要があります.

実験値入力ファイルのフォーマットを前記の例で使用した `expTest.dat` を例に説明します.
`expTest.dat` の内容は以下のようにになっています.

503	←	データ数
0	0	
0.01	0.248689883	実験データ
0.02	0.481753667	
0.03	0.684547097	
0.04	0.844327916	
...		

最初の1行目はデータ数, 2 行目からは実際の実験データを記入します.

1 列目は時間を示し, 2 列目はその時間における `effort` の値(この場合は力)を示しています.
このようなファイルを作りそれを実験値として SE または SF に設定します.

実験値入力ファイルには以下の制限があります.

- 実験値入力のデータ点数には制限があり 10000 個を超えるデータは扱えません.
- ひとつの素子に 2 つ以上の実験値は入力できません.
- 1 列目(時間)と 2 列目(`effort` または `flow`)の間には必ず空白を作ってください.
- BGSP2.0 以降で計算を行う時に, BGSP2.0 以降(本例では $1.0E-05$ [s])と実験値の時間幅(`expTest.dat` では $1.0E-02$ [s])は違う場合, 自動的にプログラム上で線形補間を行います. 実験値の時間幅をおおきくすると, 線形補間による誤差が大きくなります. また, 実験値の時間幅は BGSP2.0 以降の時間より必ず大きくしてください.

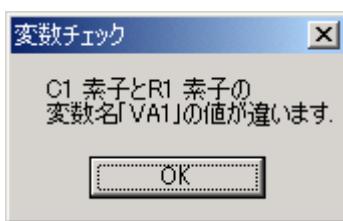
7. その他の機能

7-1 変数チェック機能

変数チェック機能は、特性式で指定した変数の誤りをチェックします。

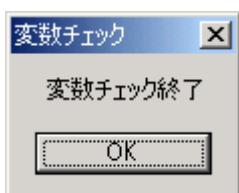
BGSP2.0 以降において変数は「各素子において同じ変数名が使われているならば、それらは同じものとして扱う」と定義しています。よって、各素子において同じ変数名を使用する場合はそれら値は同じでなければなりません。

たとえば、C1 素子に VA1, R1 素子に VA1 という変数が定義してあると仮定すると、その 2 つの変数は同じものとして扱われます。よって、もし C1 素子における VA1 の値が 1.0, R1 素子における VA1 の値が 2.0 ならば、[ツール]-[変数チェック]を実行すると以下のダイアログボックスが表示されます。



このエラーを修正するには、変数 C1 素子の VA1 の値を 2.0 にする(もしくは、R1 素子の VA1 の値を 1.0 にする)必要があります。

もし、変数に誤りがなかったら以下のダイアログボックスが表示されます。



7-2 グローバル変数 XX[i]

グローバル変数 XX[i]が用意されており、特性式間の変数の受け渡しなどに利用できます。ただし、(i=0~29)となってます。

7-2-1 再帰型ループの解消

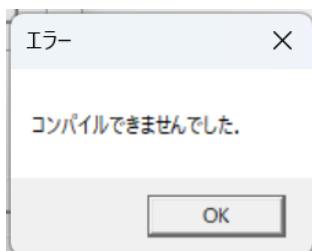
ボンドグラフで、FB 信号を利用する場合に、再帰型ループ(Recursive Loop)、あるいは再帰の問題がしばしば発生する。再帰ループとは、1つ前の Loop の Output が次の Loop の Input として帰ってくるものである。このため、連立常微分方程式を積分解法で解く場合、入力値として被積分変数がフィードバック信号として入力される場合に、再帰の問題が顕在化する。

このような場合、時間刻みの1時刻前の値をグローバル変数 XX(i)に入力してプログラムすれば再帰の問題を回避できる。通常、計算の時間刻みは 10^{-3} (sec)以下なので、僅かな時間遅れを加味したという事になる。

7-3 temp.c を利用したデバッグ作業

BGSP Ver. 2.2 より中間ファイル temp.c を確認することができるようになりました。BGSP のファイル構造は付録.4 を参照してください。

BGSP で発生するエラーは、C のコンパイル時によく発生します。その場合、以下のようなメッセージボックスが表示されます。



この場合、コマンドプロンプトを起動し、BGSP が置いてあるディレクトリに移動します。

A screenshot of a Windows Command Prompt window titled 'x64 Native Tools Command Pr'. The window shows the following text:

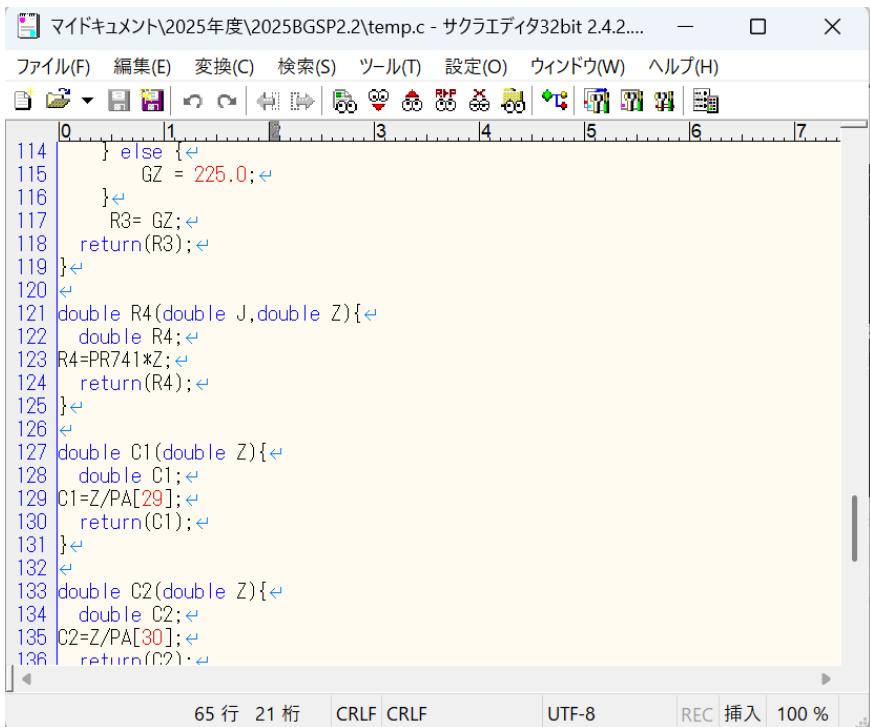
```
*****  
** Visual Studio 2022 Developer Command Prompt v17.14.16  
** Copyright (c) 2025 Microsoft Corporation  
*****  
[vcvarsall.bat] Environment initialized for: 'x64'  
  
C:\Program Files\Microsoft Visual Studio\2022\Community>cd C:\Users\hiro\Documents\2025年度\2025BGSP2.2  
  
C:\Users\hiro\Documents\2025年度\2025BGSP2.2>
```

コマンドプロンプトで bcc32c temp.c と入力してみてください。

A screenshot of a Windows Command Prompt window titled 'x64 Native Tools Command Pr'. The window shows the following text:

```
C:\Users\hiro\Documents\2025年度\2025BGSP2.2>bcc32c temp.c  
Embarcadero C++ 7.30 for Win32 Copyright (c) 2012-2017 Embarcadero Technologies, Inc.  
temp.c:  
temp.c:123:4: error: use of undeclared identifier 'PR741'  
R4=PR741*Z;  
^  
1 error generated.  
  
C:\Users\hiro\Documents\2025年度\2025BGSP2.2>
```

エラーメッセージより temp.c の 123 行目に不具合が発生していることがわかります。それでは次に temp.c の 123 行目を確認します。

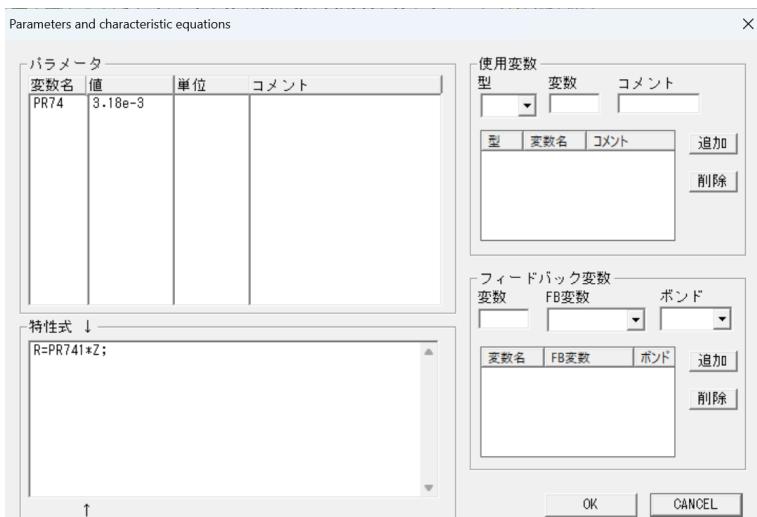


```

114 } else { ↵
115     GZ = 225.0; ↵
116 }
117     R3= GZ; ↵
118     return(R3); ↵
119 } ↵
120 ↵
121 double R4(double J,double Z){ ↵
122     double R4; ↵
123     R4=PR741*Z; ↵
124     return(R4); ↵
125 } ↵
126 ↵
127 double C1(double Z){ ↵
128     double C1; ↵
129     C1=Z/PA[29]; ↵
130     return(C1); ↵
131 } ↵
132 ↵
133 double C2(double Z){ ↵
134     double C2; ↵
135     C2=Z/PA[30]; ↵
136     return(C2); ↵

```

123 行目の PR741 が、パラメータを意味する PA 配列に置き換わっていないことがわかります。



BGSP Ver.2 の R4 の特性式入力画面を確認すると、パラメータの変数名と特性式入力の変数名が一致していないことがわかります。この変数名を一致させることでエラーを解消できます。

8. 状態方程式の導出機能

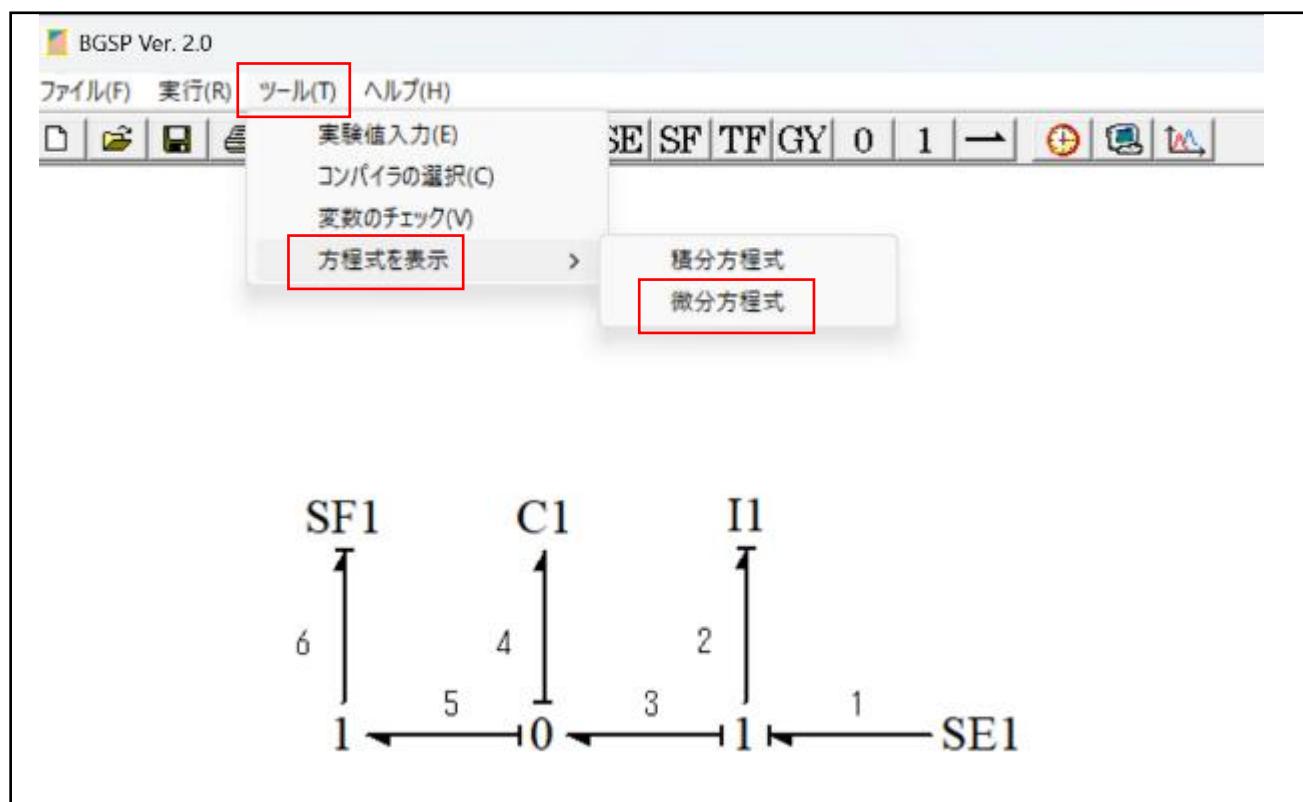
8-1 状態方程式の導出とチェック(詳細は付録. 2, 3を参照)

BGSP2.0 以降では、システム・ボンドグラフの状態方程式(支配方程式)を連立微分方程式、または連立積分方程式の形で導出することができます。

17頁に記載されているシステム・ボンドグラフと物理モデルの支配方程式(状態方程式)を次の図のように連立微分方程式の表現を求めてみましょう。

- まず、メニューバーの[ツール]から[方程式を表示]をクリックすると、[積分方程式を表示][微分方程式を表示]の2種のコマンドが現れます。どちらかを選択すると、システム・ボンドグラフの状態方程式が連立一次の形で、下図のような Window—Equation で表示されます。左図が微分形、右図が積分形表示です。
- 表示された方程式の中で使用されている文字の意味は、以下のようになっています。
付録2. 4. BG の要素表現と特性関数 を参照してください。

連立一次系の状態方程式は、C 素子と I 素子からもたらされます。 e, f はエフオート変数、フロード変数、 C と L は C 素子と I 素子の定義式の係数(直線運動系であれば、 $L = 1/M$ (M :質量)、 $C = K$ (ばね定数))に相当し、添え字の数字はボンド番号、記号の頭につけられているドット記号・は微分をそれぞれ意味します。また、 $E(t), F(t)$ は境界条件としての SE 素子と SF 素子の特性値が入ります。



Equation

$$\dot{f}_2 = -L_1 e_4 + L_1 E_1(t)$$

$$\dot{e}_4 = -C_1 F_1(t) + C_1 f_2$$

$$\frac{dv_1}{dt} = -\frac{1}{M} e_4 + \frac{1}{M} E_1$$

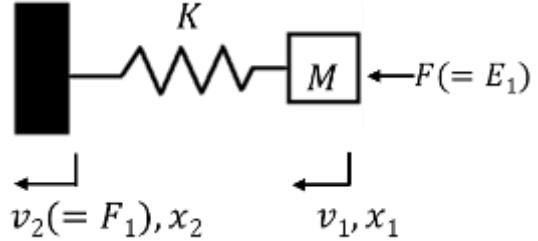
$$\frac{de_4}{dt} = -KF_1 + Kv_1 = K(v_1 - v_2)$$

$$\longrightarrow e_4 = K(x_1 - x_2)$$

Equation

$$e_2 = -C_1 \int f_4 dt + E_1(t)$$

$$f_4 = -F_1(t) + L_1 \int e_2 dt$$



これを上式に代入すると,

$$M \frac{dv_1}{dt} = E_1 - K(x_1 - x_2)$$

9. BGSP2.0以降における制限

9-1 素子数, ボンド数, パラメータ数の制限

素子数の制限は以下のようになっています.

○ C version の場合

使用可能素子数 500 個まで
使用可能ボンド数 500 個まで
パラメータ数 999 個まで

○ Fortran version の場合

使用可能素子数 500 個まで

注) 素子の種類によって制限があります.

SE, SF 素子	あわせて 400 個まで
C, I 素子	あわせて 400 個まで
R 素子	400 個まで
TF, GY 素子	あわせて 400 個まで
0, 1 接点	あわせて 400 個まで
使用可能ボンド数	500 個まで
パラメータ数	999 個まで

9-2 使用禁止変数

以下に示す変数名は使用できません.

DE, DE, DX, ERR, ERRNUM, fp1, fp2, H, ING, ITC, LABEL, ND, NOT
NOUT, NS, NT, OP, PA, PT, T, TO, T1, TI, X, XX, ODD, QUOTIENT, CONSTVALUE
CFILE, DOUT, DSIGN, FEHL, FU, FUNC, ICHEK, INDEX, INIT, main, OFILE
PARM, RG2, RUNGE, SOLV

使用素子名(C1, C2, SE1 など)

9-3 その他の制限

○ 1 つの素子あたりの制限

- ・特性式に入力可能な文字数 1000 文字まで
- ・フィードバック変数 20 個まで
- ・パラメータ数 16 個まで
- ・使用変数の数 20 個まで
- ・素子に接続可能なボンド数 8 個まで

○ 計算結果を出力できる変数数 100 個まで

10. 例 1 マス・ばね・ダンパーにおけるボンドグラフ

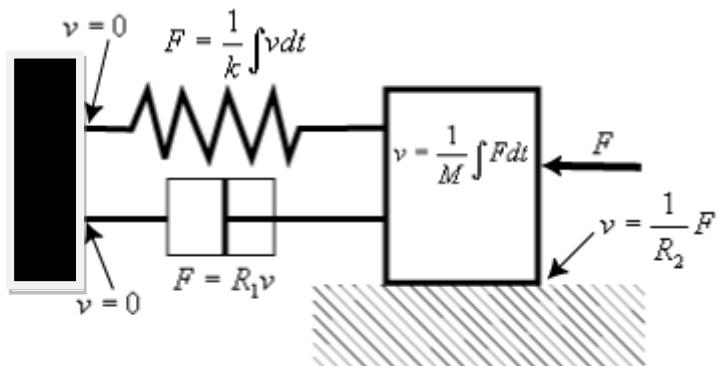


図 10-1 マス・ばね・ダンパー モデル

図 10-1 は 1 自由度振動系物理モデルです. F は力, v は速度, t は時間, M は質量, R_1 はダンパー係数, R_2 は摩擦係数, k はばね係数を表します.

図 10-1 の物理モデルをボンドグラフにすると図 10-2 のようになります.

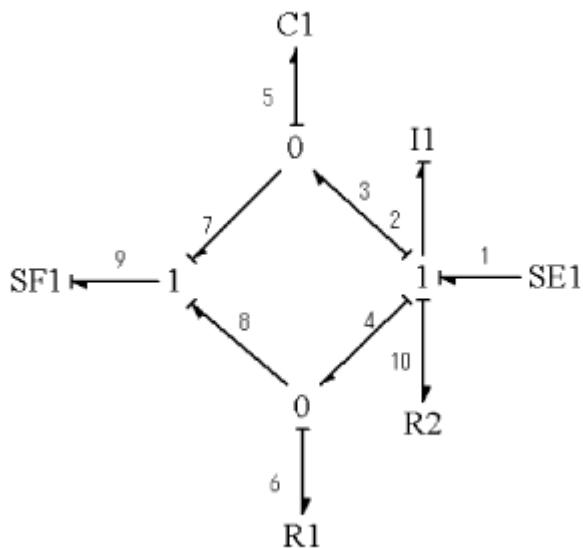


図 10-2 ボンドグラフ モデル

ここで, $SE1$ は外力, $I1$ は慣性力, $R1$ はダンパー力, $R2$ は床との摩擦力, $C1$ はばね力, $SF1$ は壁面速度に対応しています. それぞれの入力パラメータを次に示します.

10-1 C version の場合

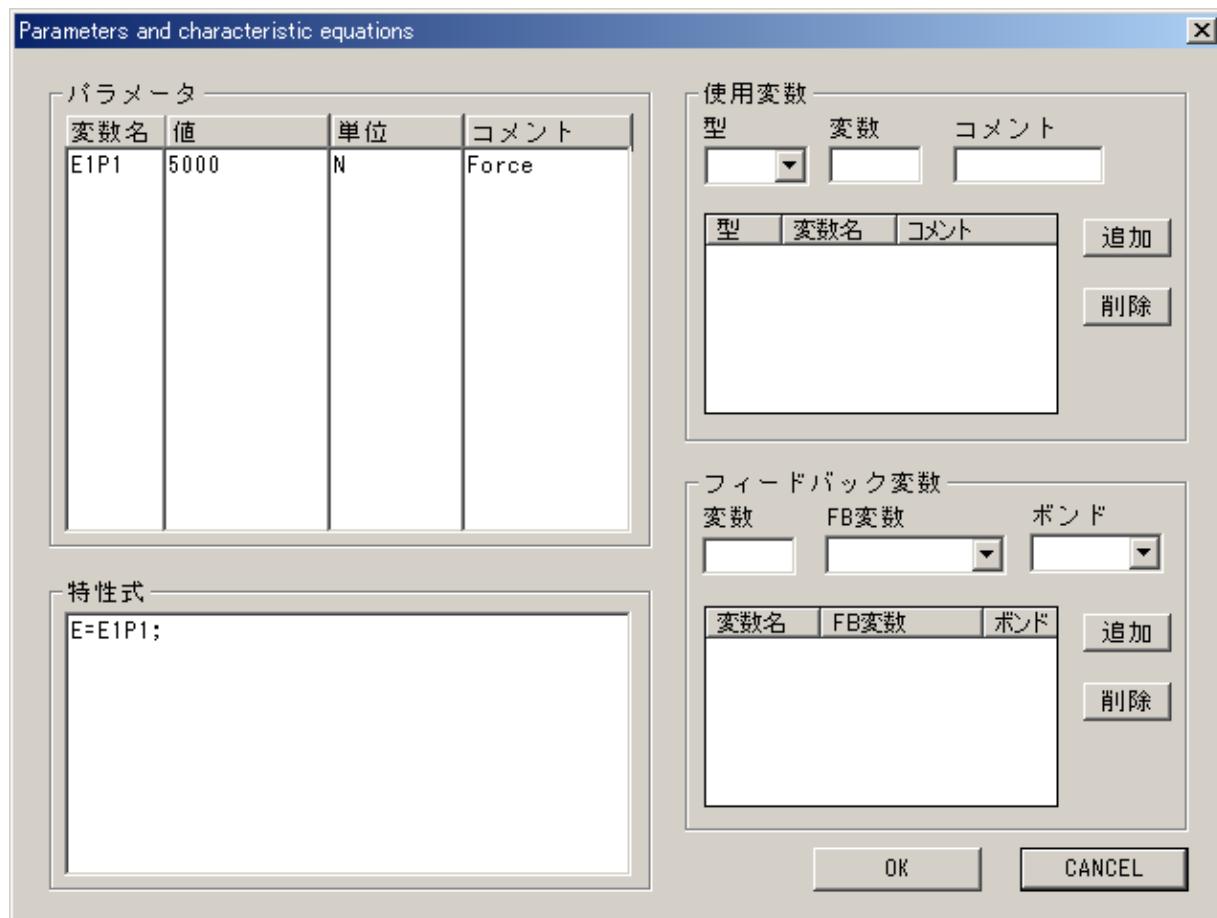
表 10-1に入力パラメータ情報を示します。

表 10-1 C version における入力パラメータ情報

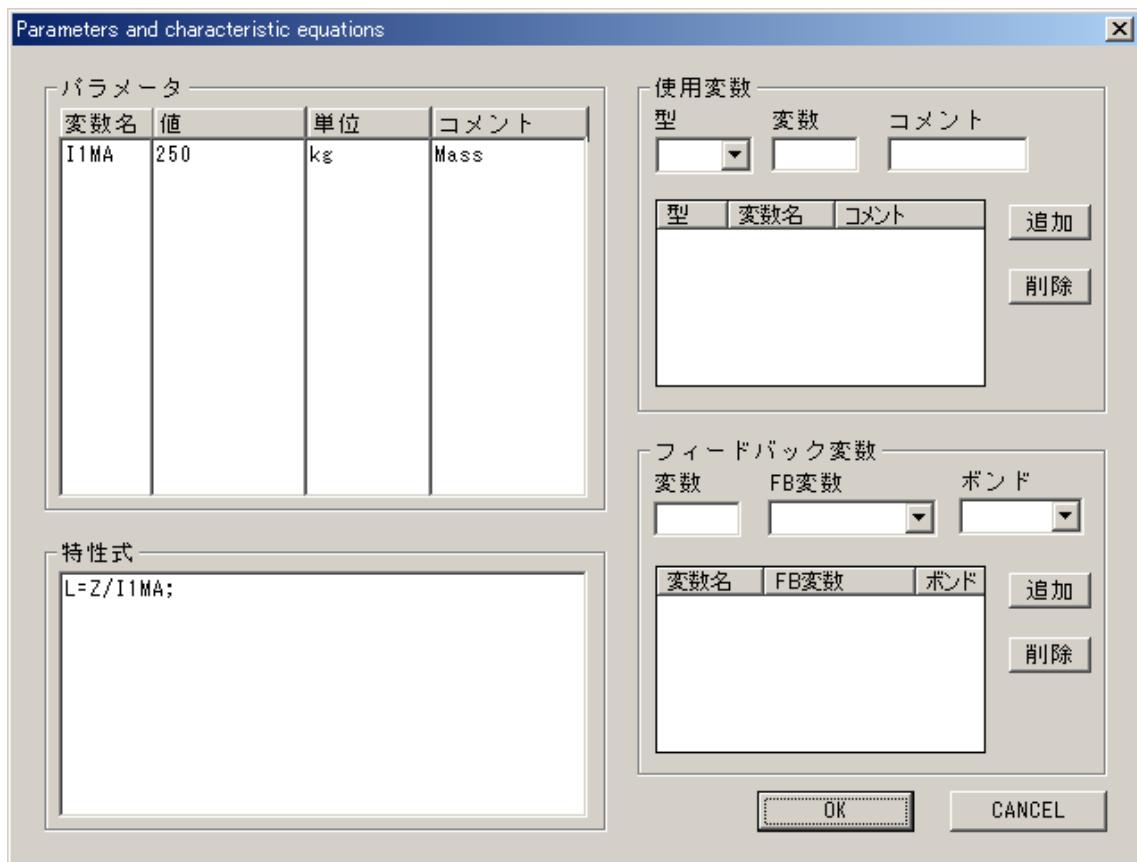
素子名	変数名	値	単位	コメント	特性式
SE1	E1P1	5000	[N]	Force	$E=E1P1;$
I1	I1MA	250	[kg]	Mass	$L=Z/I1MA;$
C1	C1SP	9800	[N/m]	Spring coefficient	$C=C1SP*Z;$
R1	R1DA	1000	[Ns/m]	Damper coefficient	$R=R1DA*Z;$
SF1	F1VE	0.0	[m/s]	Wall	$F=F1VE;$
R2	R2FR	100	[Ns/m]	Friction	$R=R2FR*Z;$

次に各素子の入力パラメータ情報入力を示します。

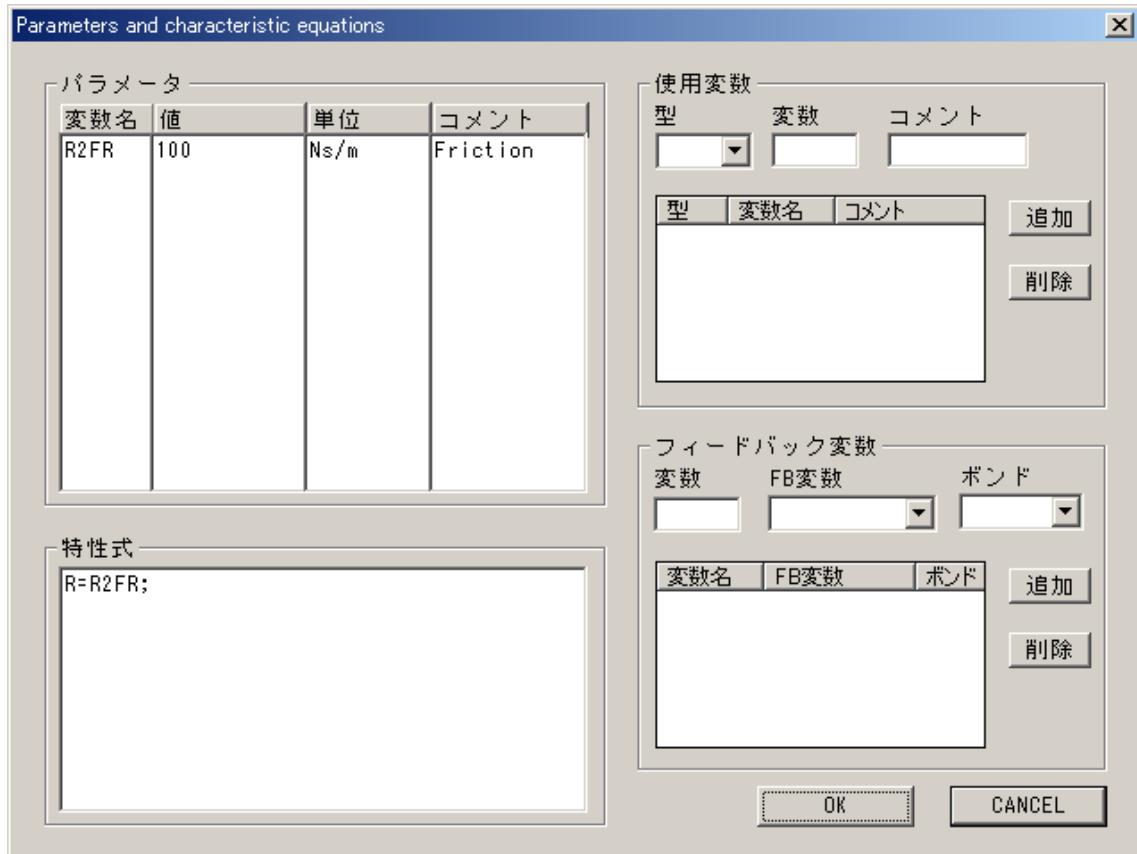
・素子 SE1



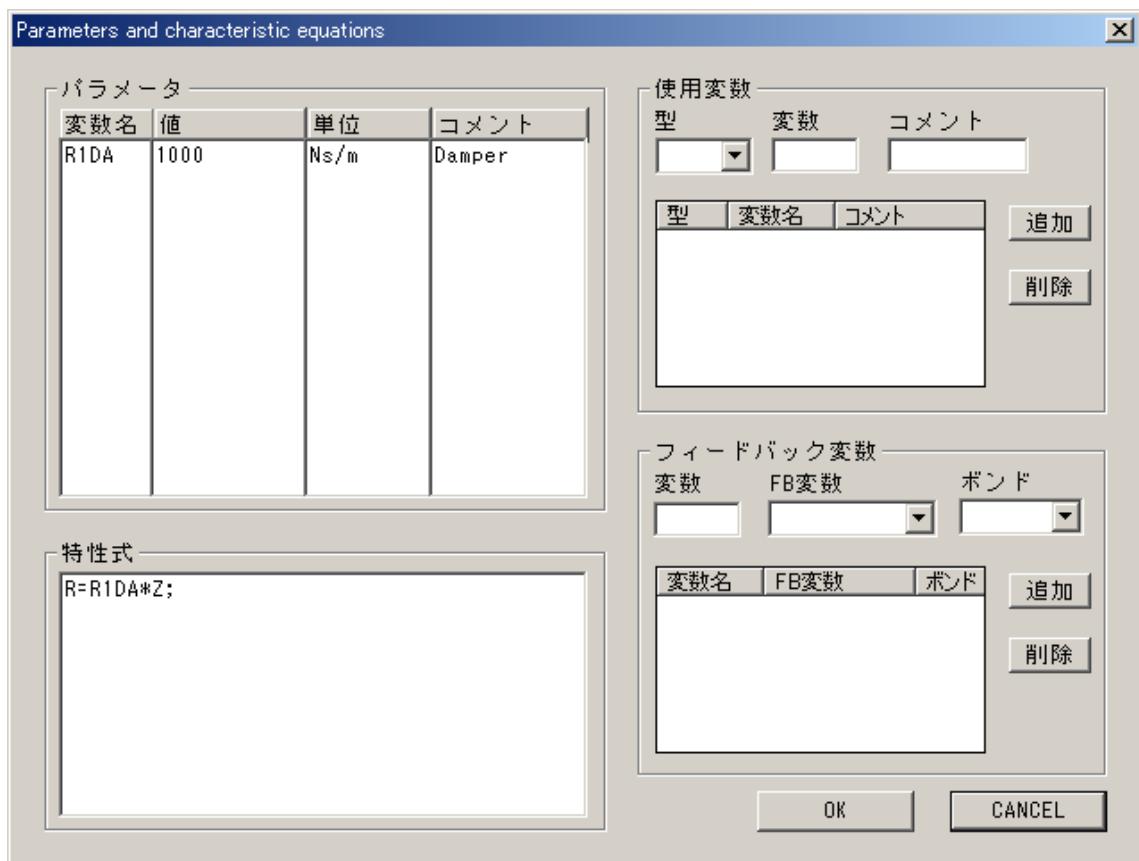
・素子 I1



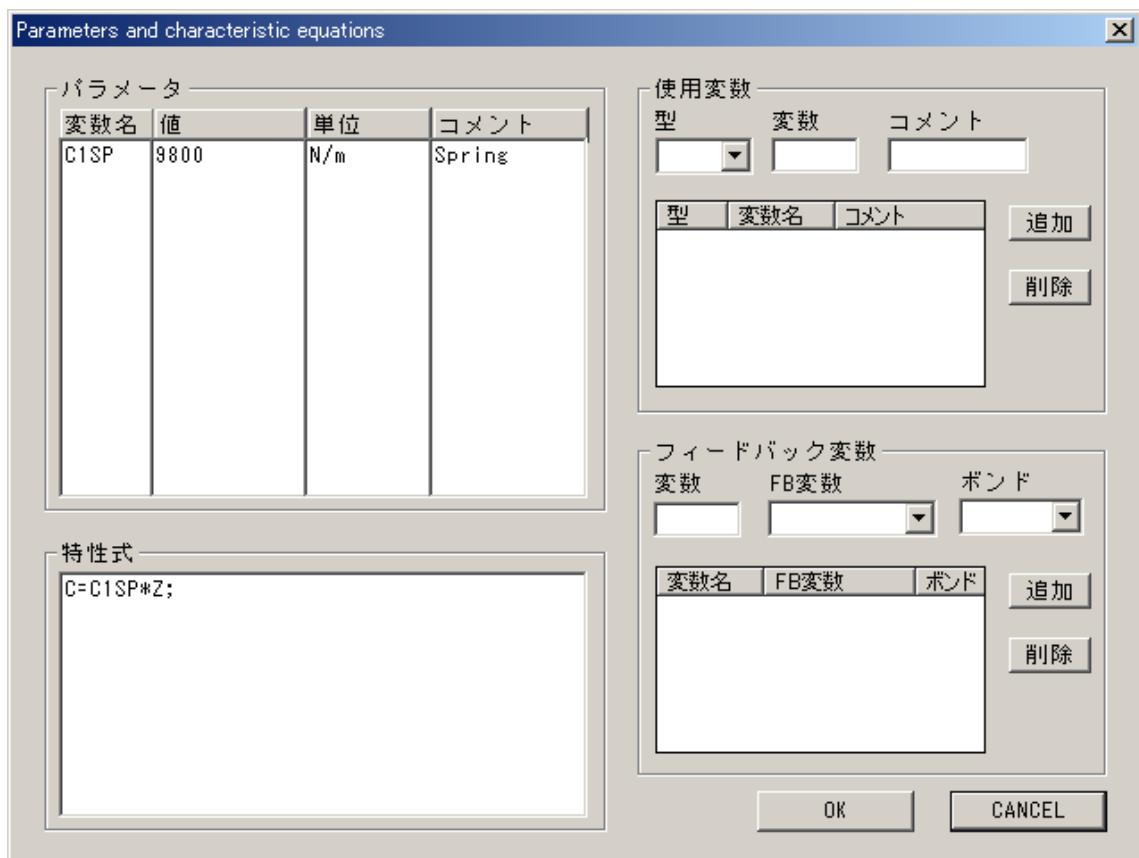
・素子 R2



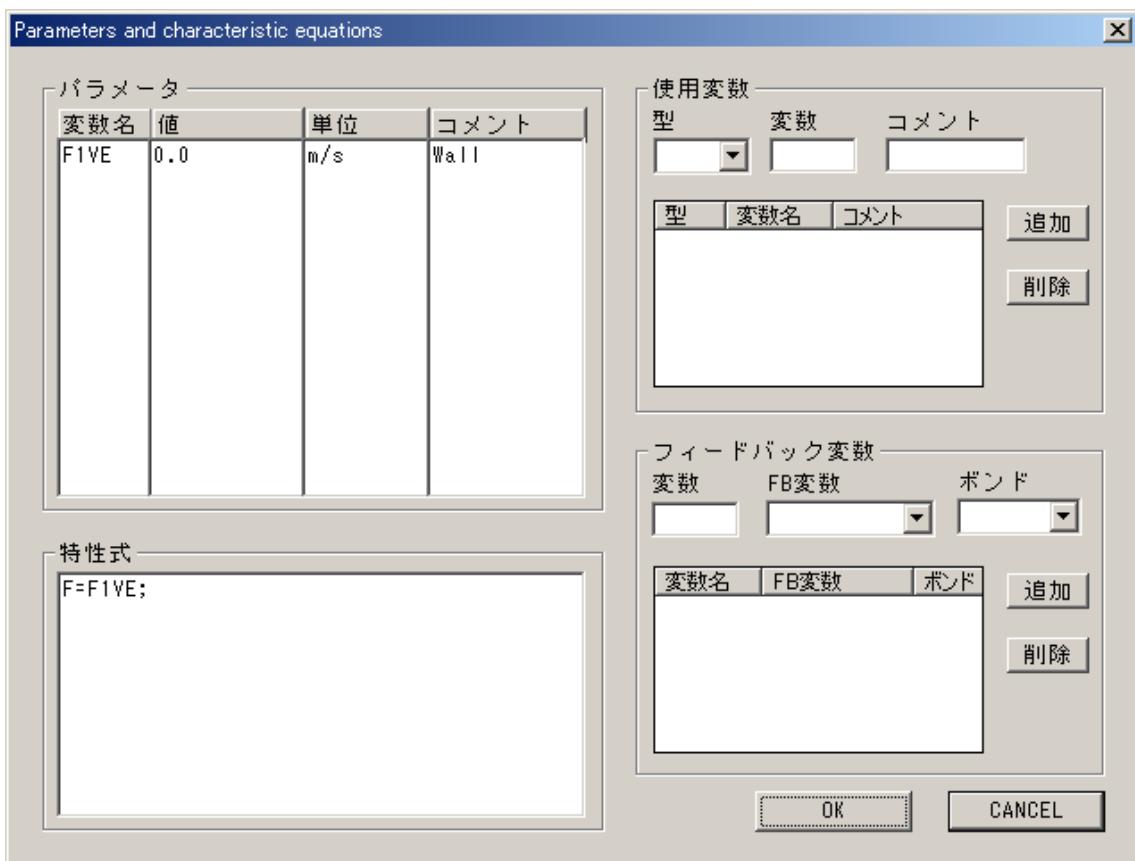
・素子 R1



・素子 C1



・素子 SF1



10-2 Fortran version の場合

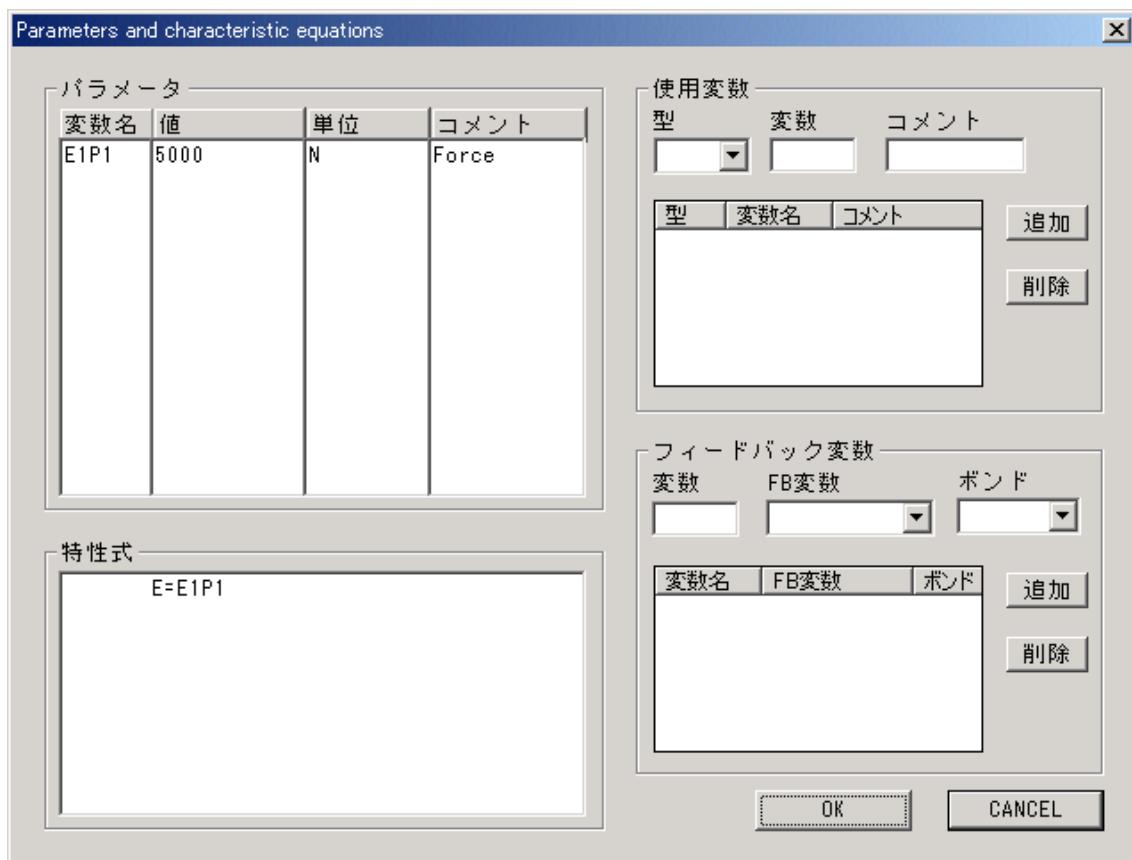
表 10-2 に入力パラメータ情報を示します。

表 10-2 Fortran version における入力パラメータ情報

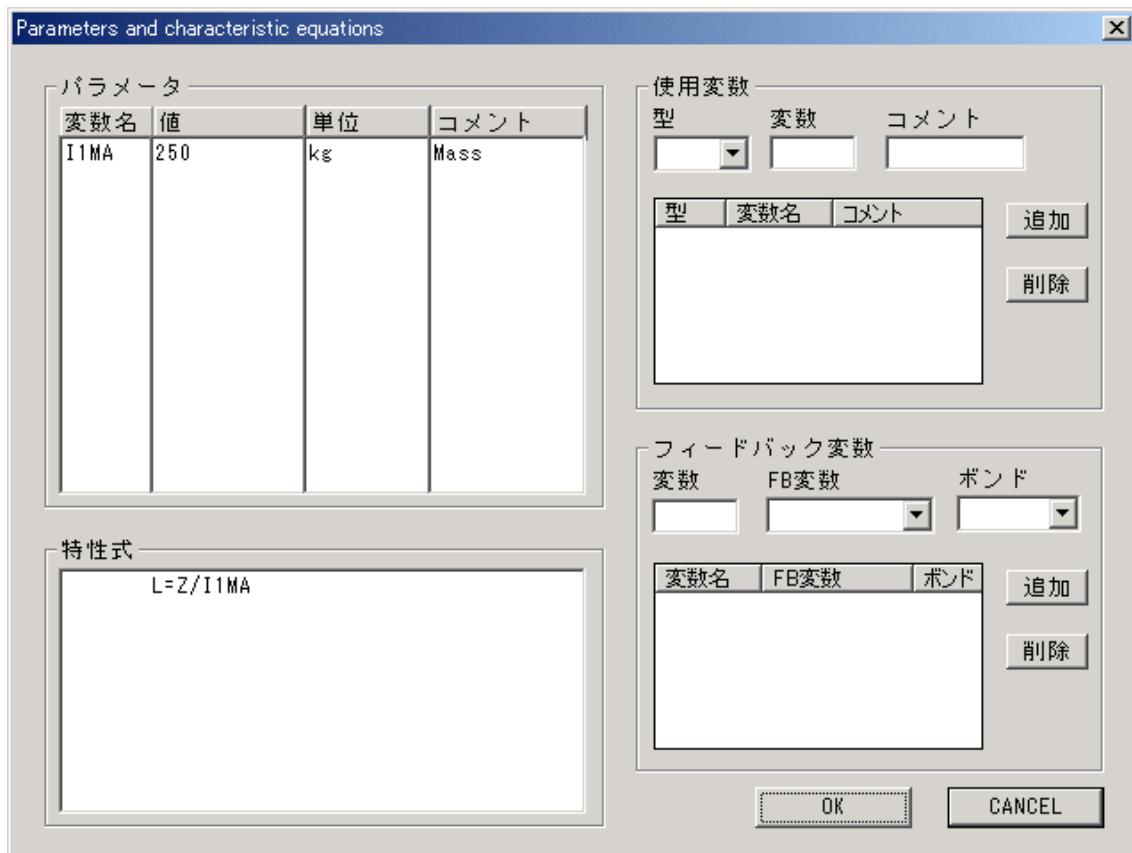
素子名	変数名	値	単位	コメント	特性式
SE1	E1P1	5000	[N]	Force	$E=E1P$
I1	I1MA	250	[kg]	Mass	$L=Z/I1MA$
C1	C1SP	9800	[N/m]	Spring coefficient	$C=C1SP*Z$
R1	R1DA	1000	[Ns/m]	Damper coefficient	$R=R1DA*Z$
SF1	F1VE	0.0	[m/s]	Wall	$F=F1VE$
R2	R2FR	100	[Ns/m]	Friction	$R=R2FR*Z$

次に各素子の入力パラメータ情報を示します。

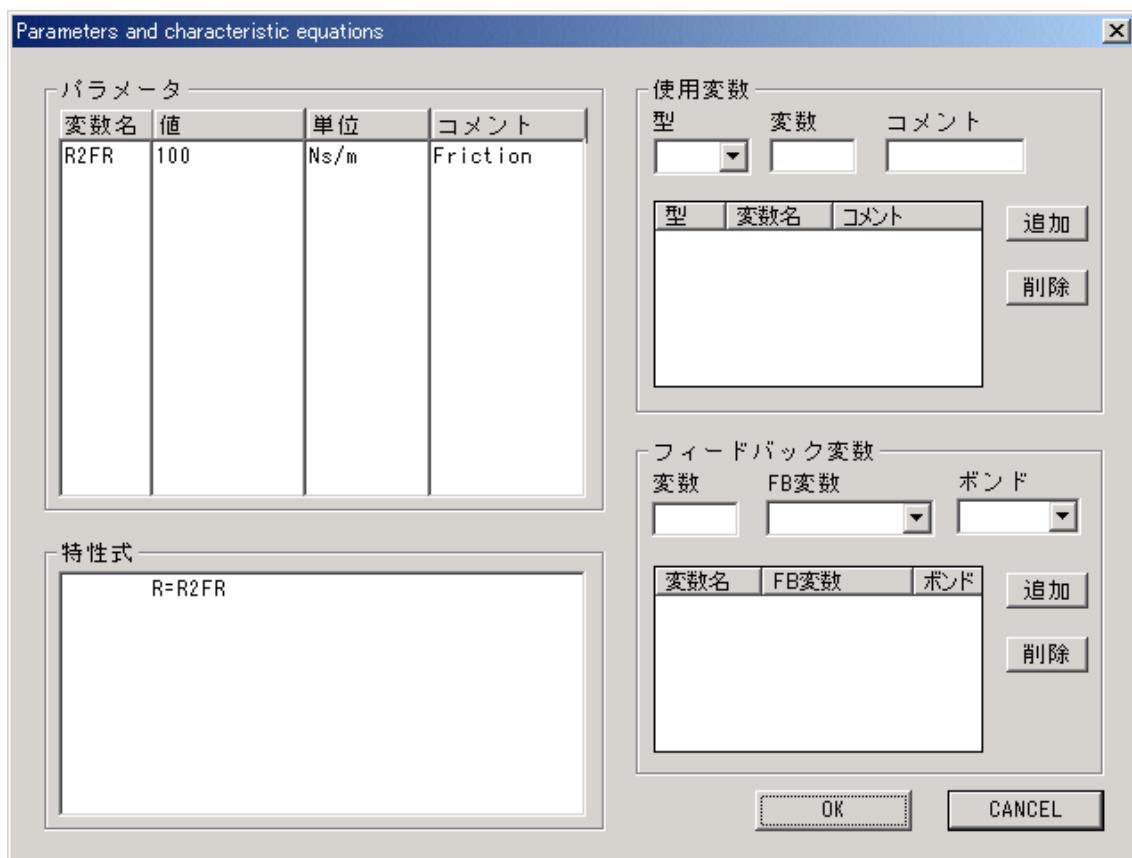
・素子 SE1



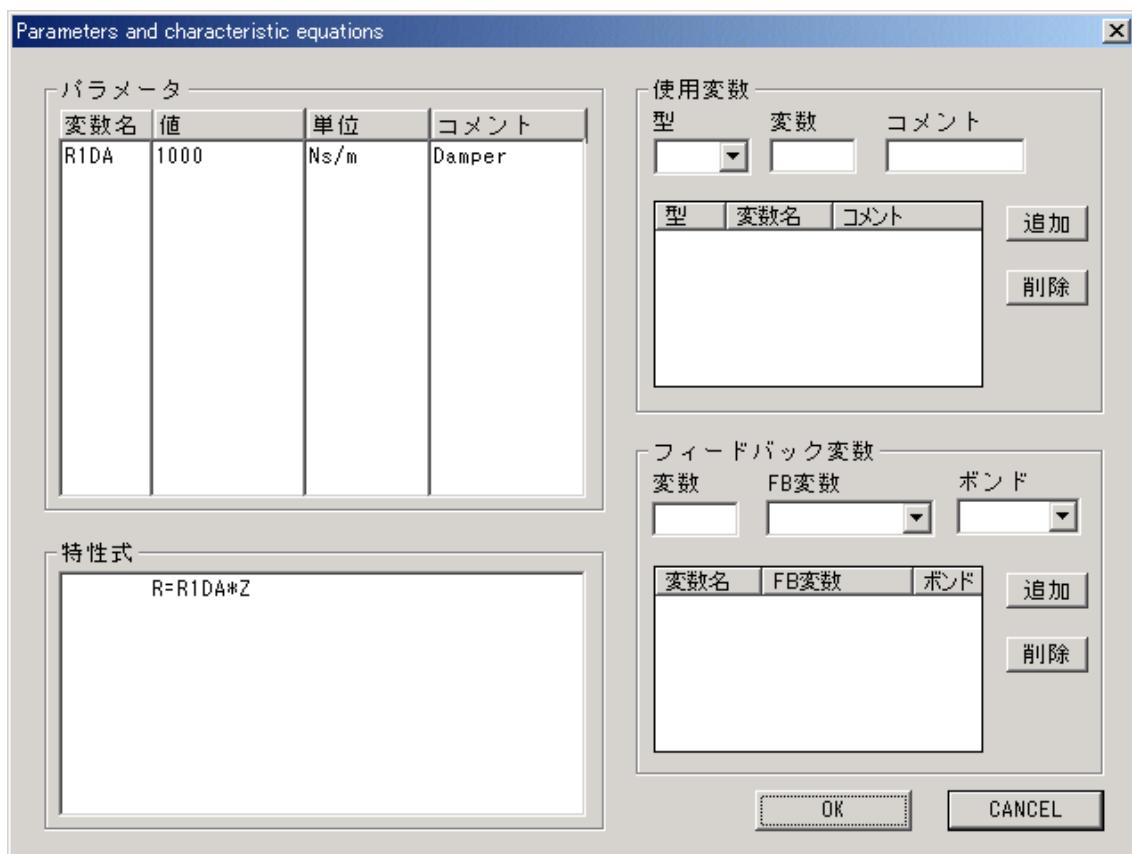
・素子 I1



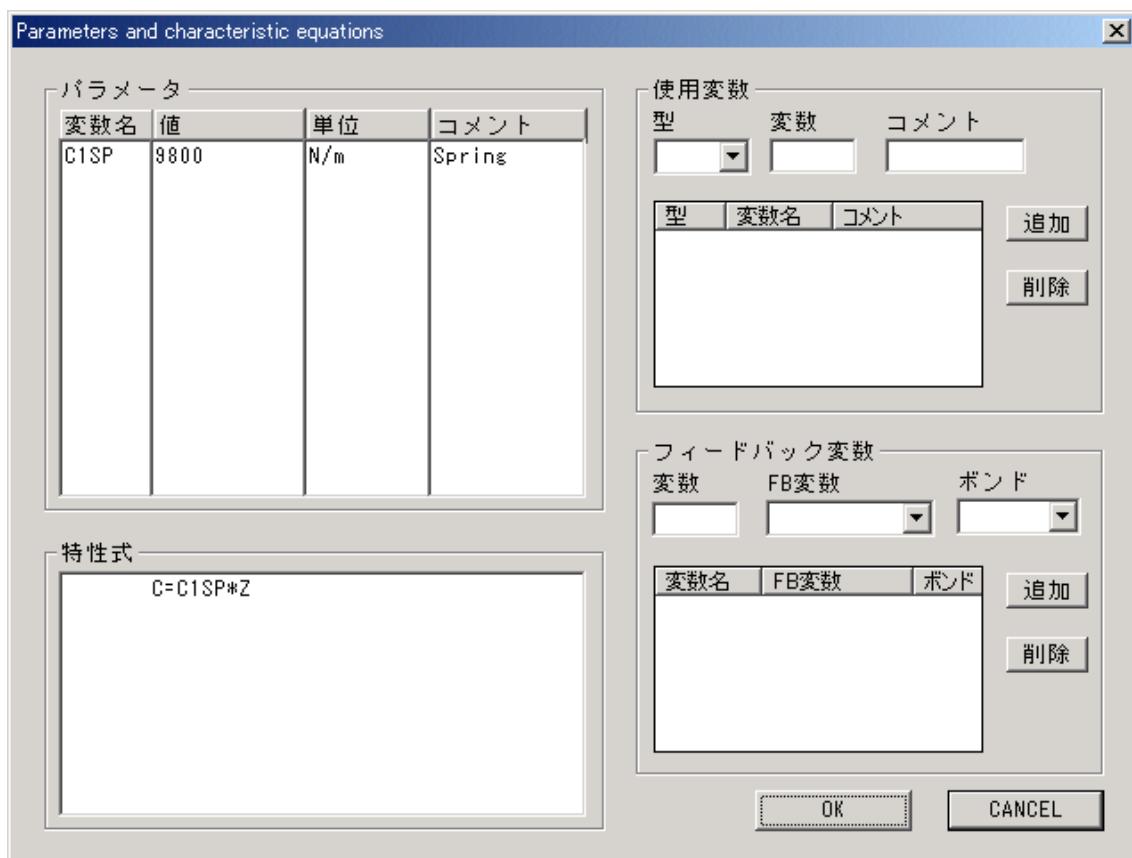
・素子 R2



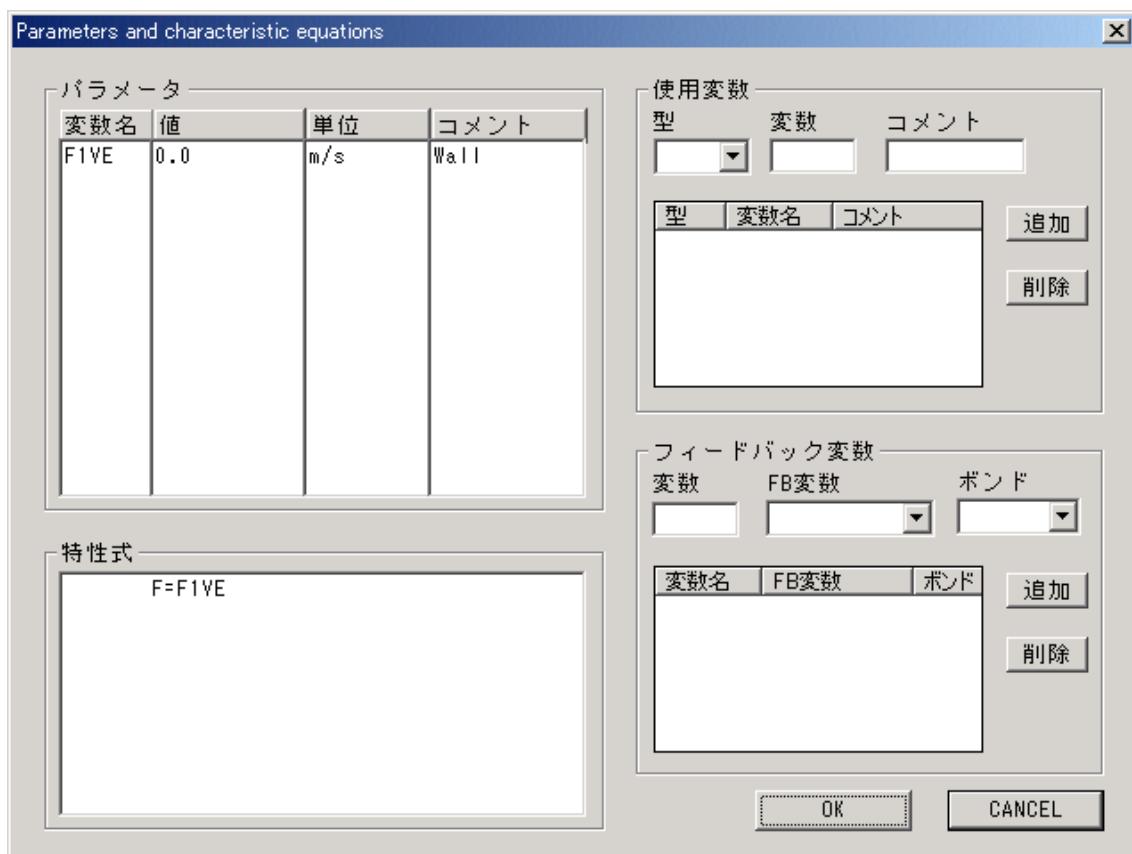
・素子 R1



・素子 C1



・素子 SF1



10-3 計算

計算パラメータを図 10-3 のように入力します。

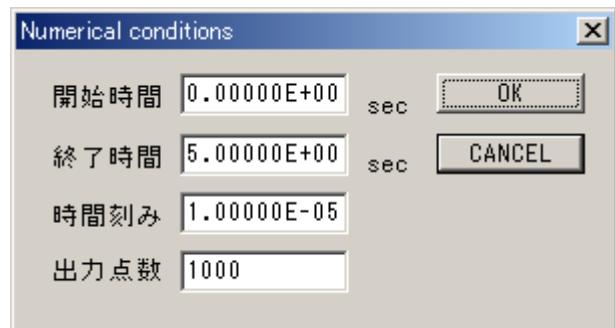


図 10-3 計算パラメータ

図 10-3 のパラメータで計算するとボンド番号2の DP は、図 10-4 のような結果になります。

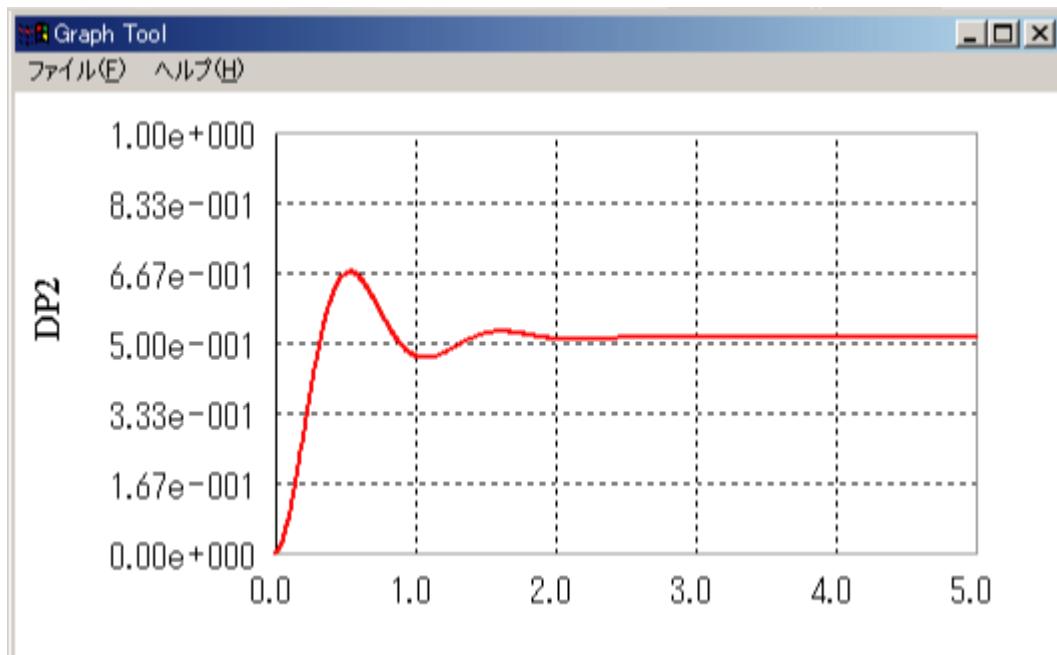


図 10-4 ボンド番号 2 の DP

11. 例 2 油圧シリンダーシステムにおけるボンドグラフ

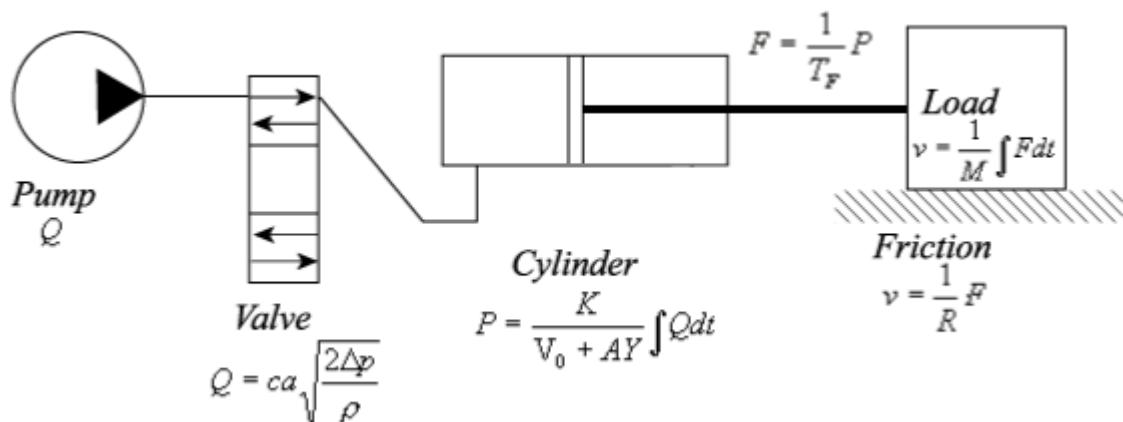


図 11-1 油圧シリンダーシステム

図 11-1 は油圧シリンダーシステムです. Q は流量, P は圧力, ρ は密度, Δp は圧力差, c 流量係数, a はバルブ流路の断面積, V_0 はシリンダーの容積, K は体積弾性率, A はシリンダーの断面積, Y はシリンダーの位置, T_F はシリンダー断面積の逆数, v は速度, M は質量, F は力, R は床との摩擦係数, t は時間を表します.

図 11-1 の物理モデルをボンドグラフにすると図 11-2 のようになります.

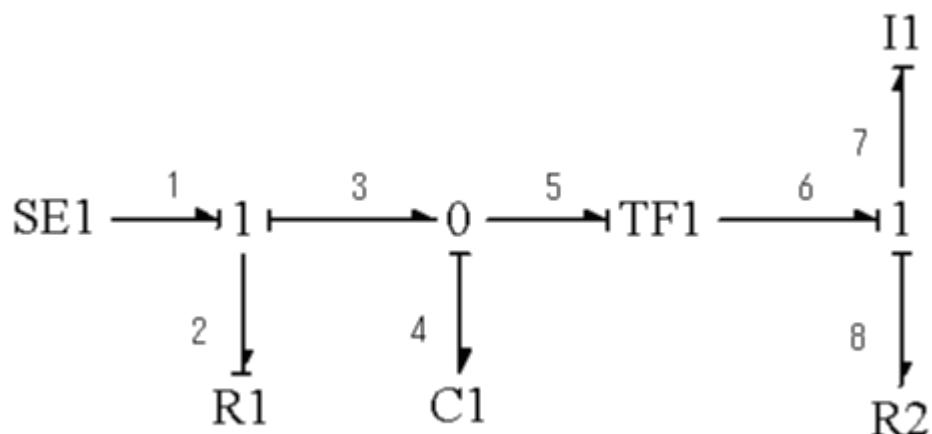


図 11-2 ボンドグラフモデル

ここで, SE1 はポンプ圧力, R1 はバルブ圧力損失, C1 はシリンダー圧力, TF1 はピストン断面積, I1 は質量慣性力, R2 は床との摩擦力を表しています.

それぞれの, 入力パラメータを次に示します.

11-1 C version の場合

表 11-1 に入力パラメータ情報を示します。

表 11-1 C version における入力パラメータ情報

素子名	変数名	値	単位	コメント
SE1	E1P1	5.0e+06	Pa	Pressure
R1	R1C	0.62	-	Flow Coefficient
R1	R1A	5.0e-05	m ²	Area
R1	R1RHO	850	kg/m ³	Density
R1	R1T	0.1	sec	Opening
C1	C1K	1.7e+09	Pa	Bulk Modulus
C1	C1D	1.0e-01	m	Diameter
C1	C1LEN	2.0e-01	m	Length
C1	C1PI	3.141592654	-	Pai
TF1	FTD	1.0e-01	m	Diameter
TF1	FTPI	3.141592654	-	Pai
R2	R2K	5.0E+03	Ns/m	Friction
I1	I1M	1.0e+04	kg	Mass

次に特性式及びフィードバック変数を示します。

SE1

特性式 $E=E1P1;$

R1

```
特性式 if(T<=R1T){  
    R=R1C*R1A*sqrt(2*Z/R1RHO)*T/R1T;  
}else{  
    R=R1C*R1A*sqrt(2*Z/R1RHO);  
}
```

C1

```
特性式 double C1A;  
C1A=C1PI*C1D*C1D/4;  
C=C1K/(C1A*C1LEN+C1A*C1DP)*Z;
```

フィードバック変数

変数名 C1DP
FB 变数 Displacement
ボンド番号 7

TF1

特性式 $TF=1/(FTD*FTD*FTPI/4);$

R2

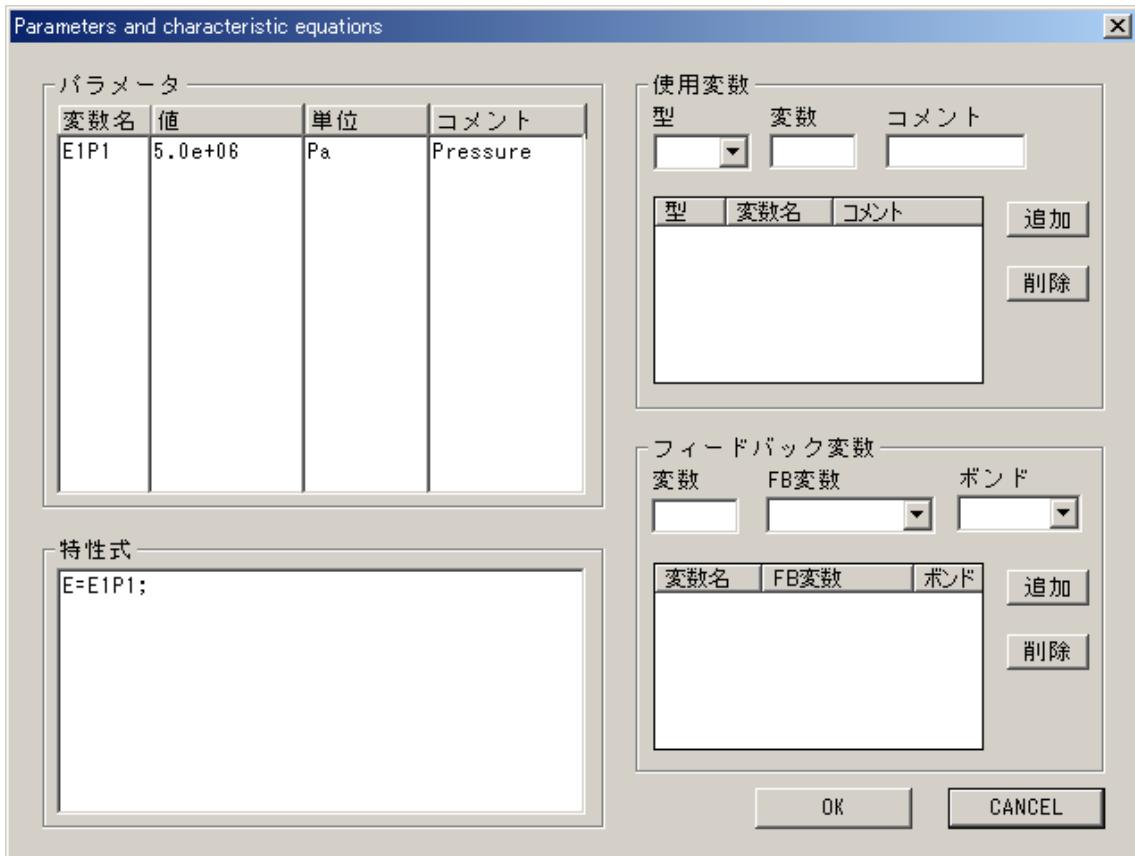
特性式 $R=R2K*Z;$

I1

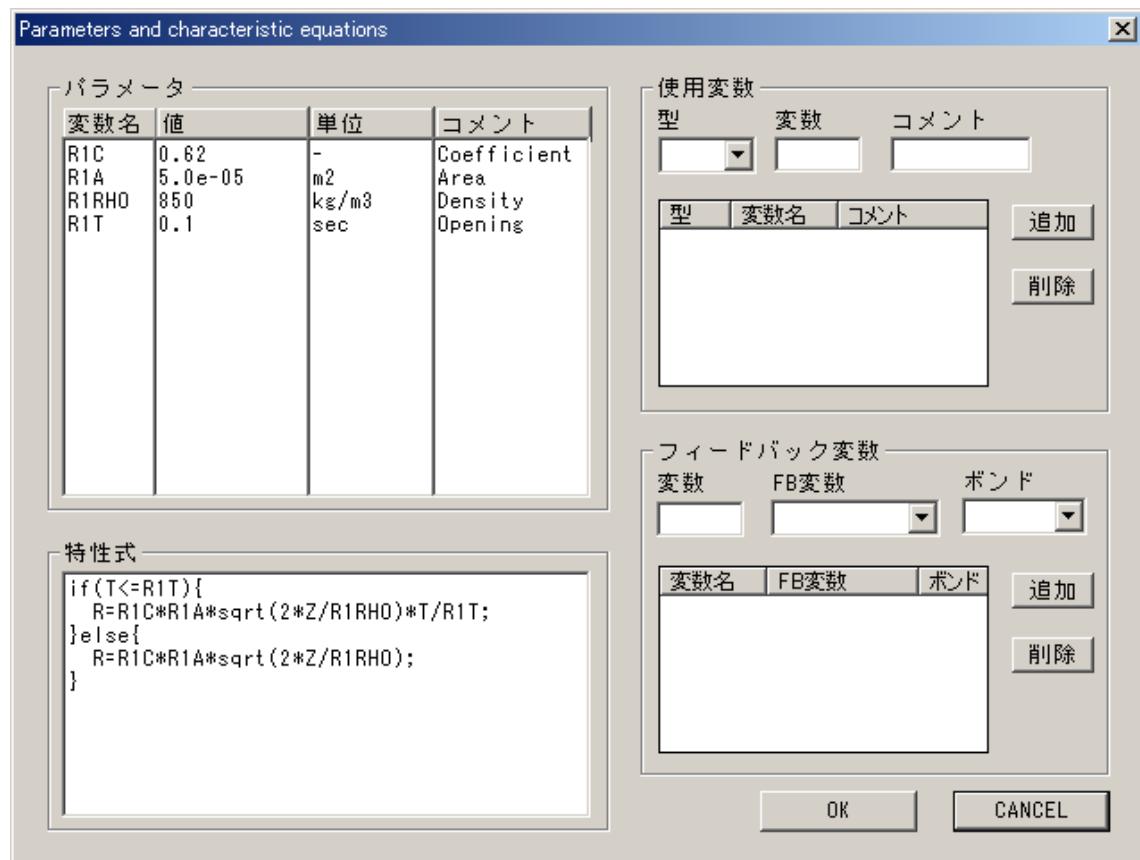
特性式 $L=Z/I1M;$

各素子の入力パラメータ情報の入力を示します.

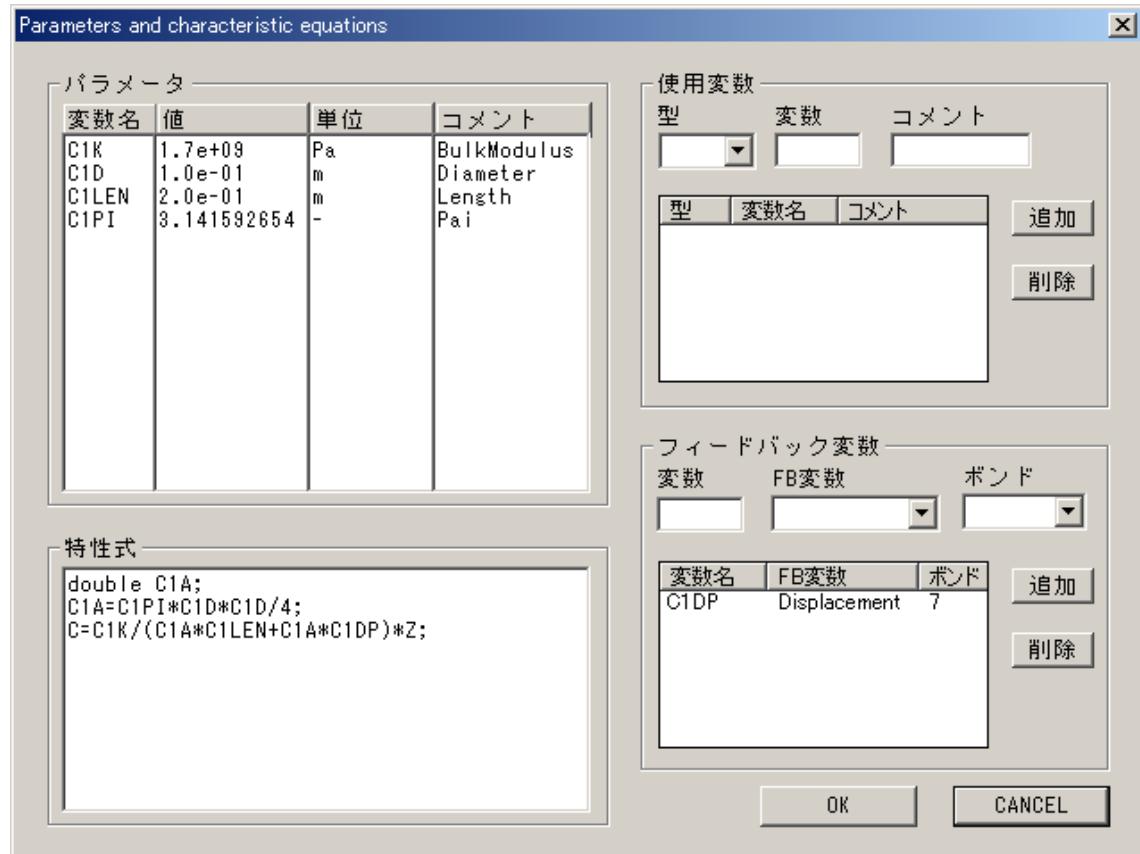
素子 SE1



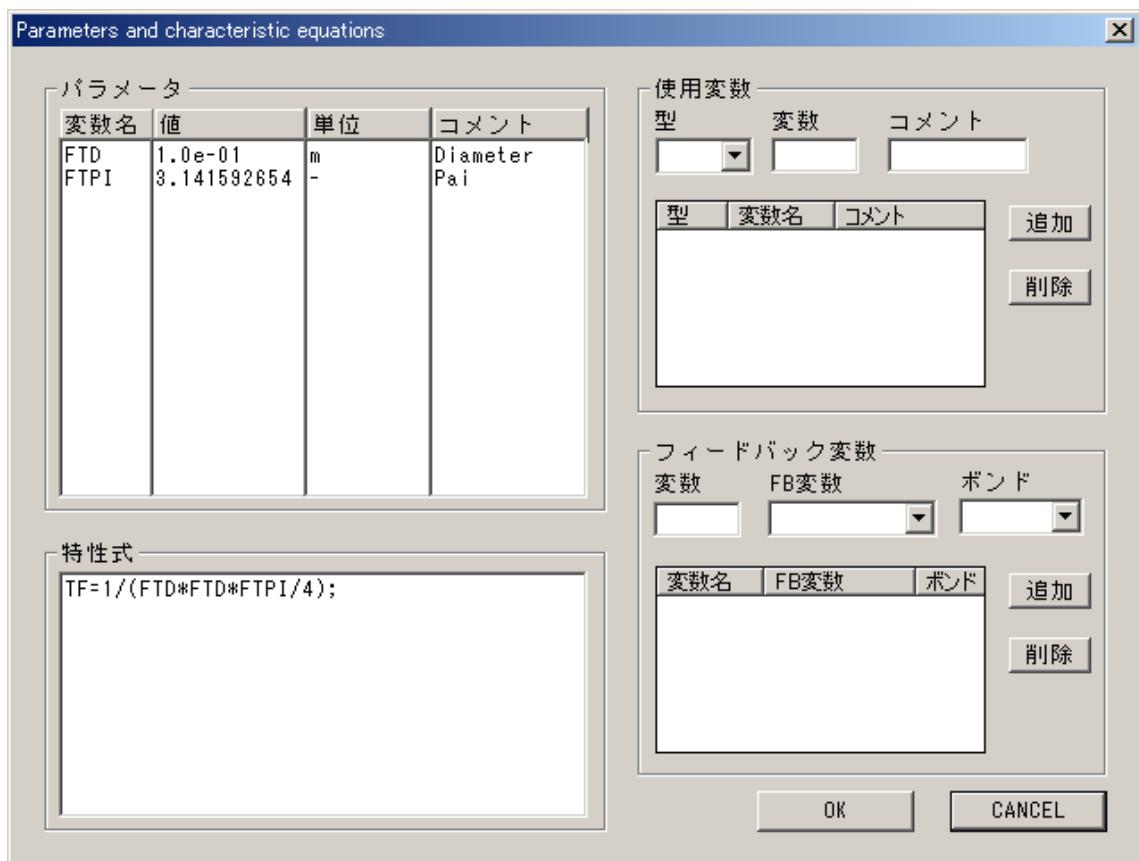
素子 R1



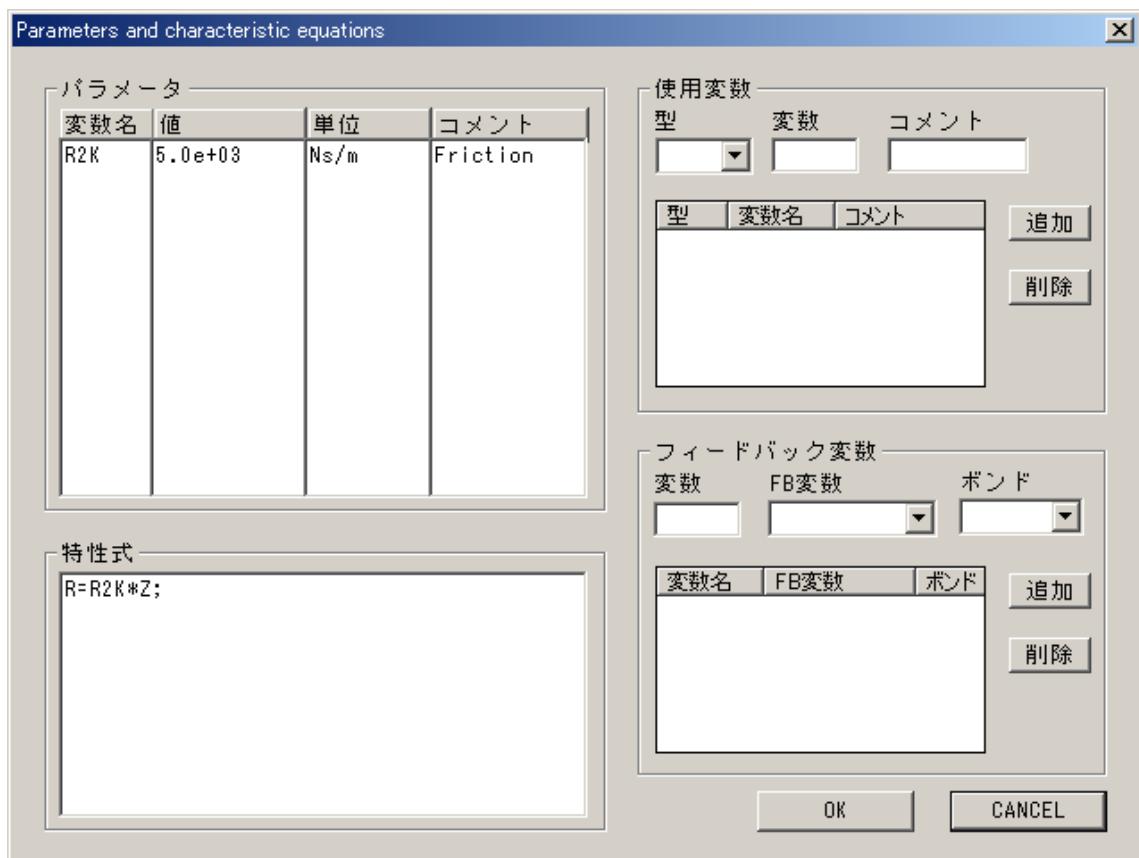
素子 C1



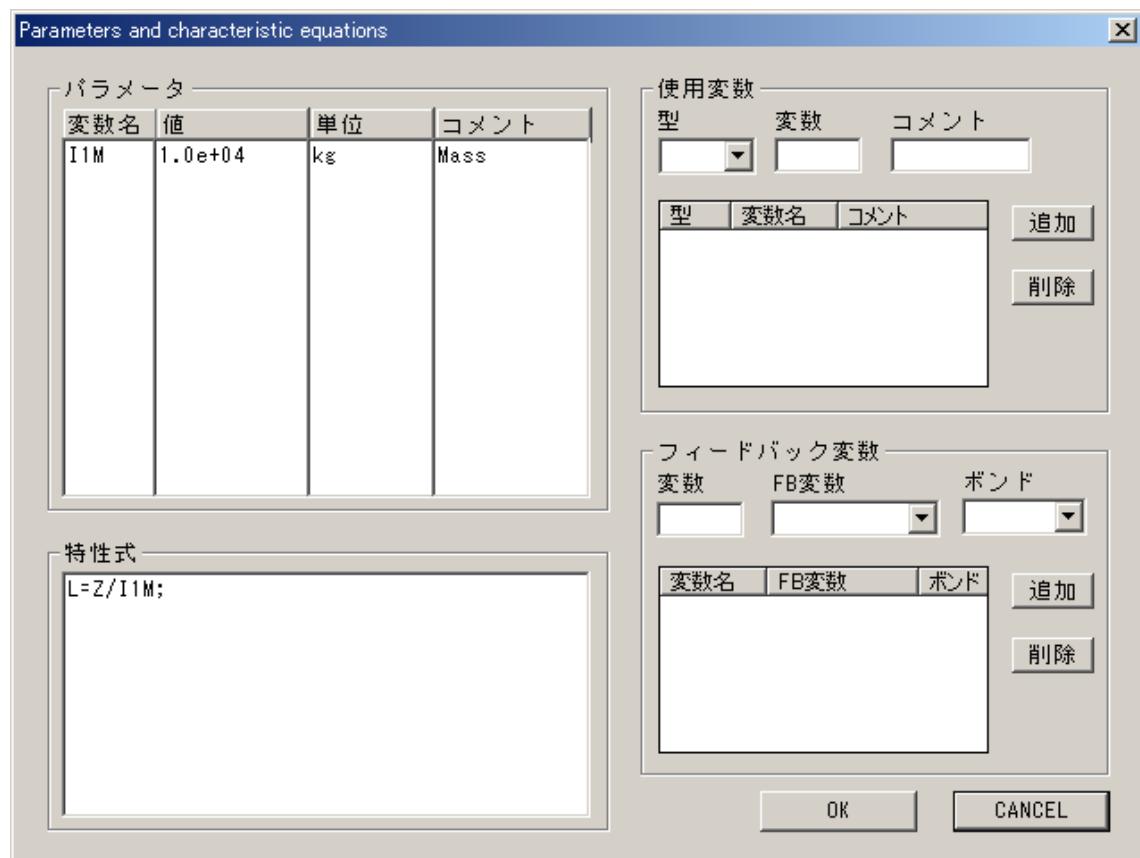
素子 TF1



素子 R2



素子 I1



11-2 Fortran version の場合

表 11-2 に入力パラメータ情報を示します。

表 11-2 Fortran version における入力パラメータ情報

素子名	変数名	値	単位	コメント
SE1	E1P1	5.0e+06	Pa	Pressure
R1	R1C	0.62	-	Flow Coefficient
R1	R1A	5.0e-05	m ²	Area
R1	R1RHO	850.0	kg/m ³	Density
R1	R1T	0.1	sec	Opening
C1	C1K	1.7e+09	Pa	Bulk Modulus
C1	C1D	1.0e-01	m	Diameter
C1	C1LEN	2.0e-01	m	Length
C1	C1PI	3.141592654	-	Pai
TF1	FTD	1.0e-01	m	Diameter
TF1	FTPI	3.141592654	-	Pai
R2	R2K	5.0E+03	Ns/m	Friction
I1	I1M	1.0e+04	kg	Mass

以下に, Fortran version における特性式及びフィードバック変数を示します.

SE1

特性式 $E=E1P1$

R1

```
特性式 IF(T.LE.R1T) THEN
    R=R1C*R1A*sqrt(2*Z/R1RHO)*T/R1T
ELSE
    R=R1C*R1A*sqrt(2*Z/R1RHO)
END IF
```

C1

```
特性式 C1A=C1PI*C1D*C1D/4
C=C1K/(C1A*C1LEN+C1A*C1DP)*Z
```

フィードバック変数

変数名 C1DP
FB 変数 Displacement
ボンド番号 7

TF1

特性式 $TF=1/(FTD*FTD*FTPI/4)$

R2

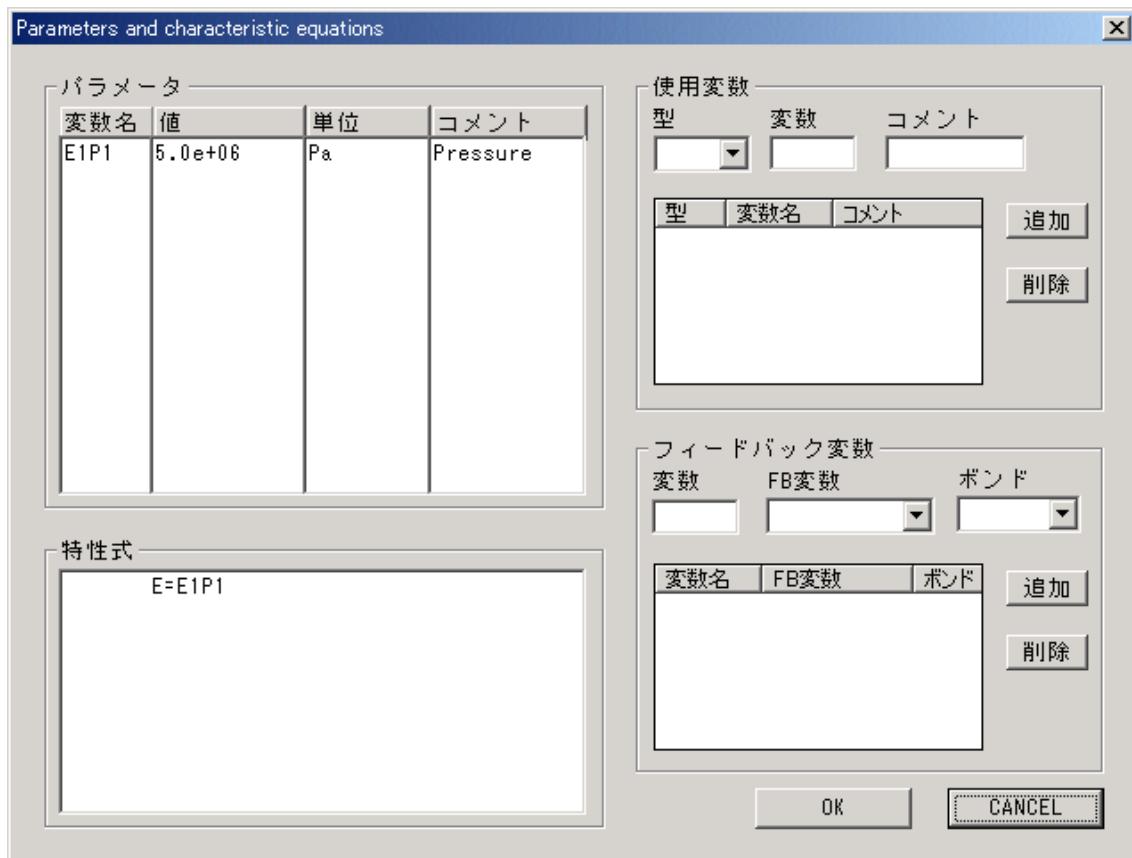
特性式 $R=R2K*Z$

I1

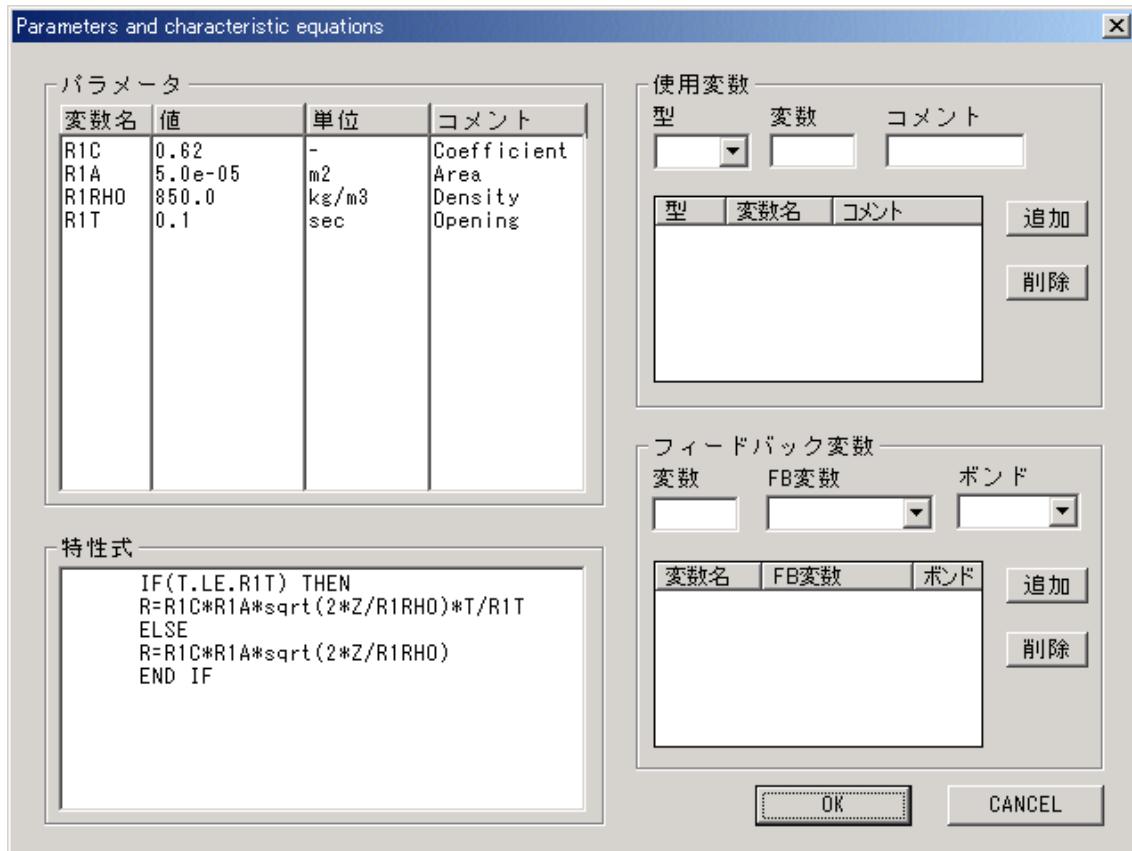
特性式 $L=Z/I1M$

次に各素子の入力パラメータ情報の入力を示します.

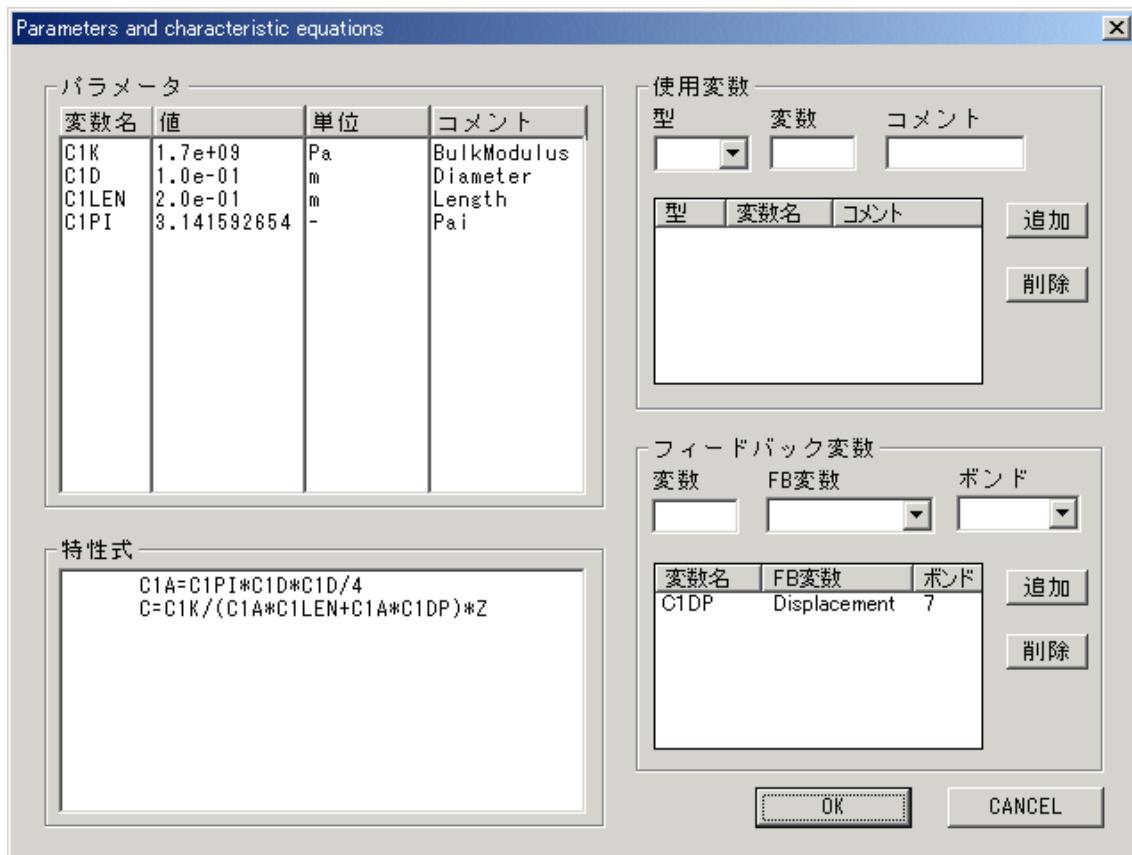
素子 SE1



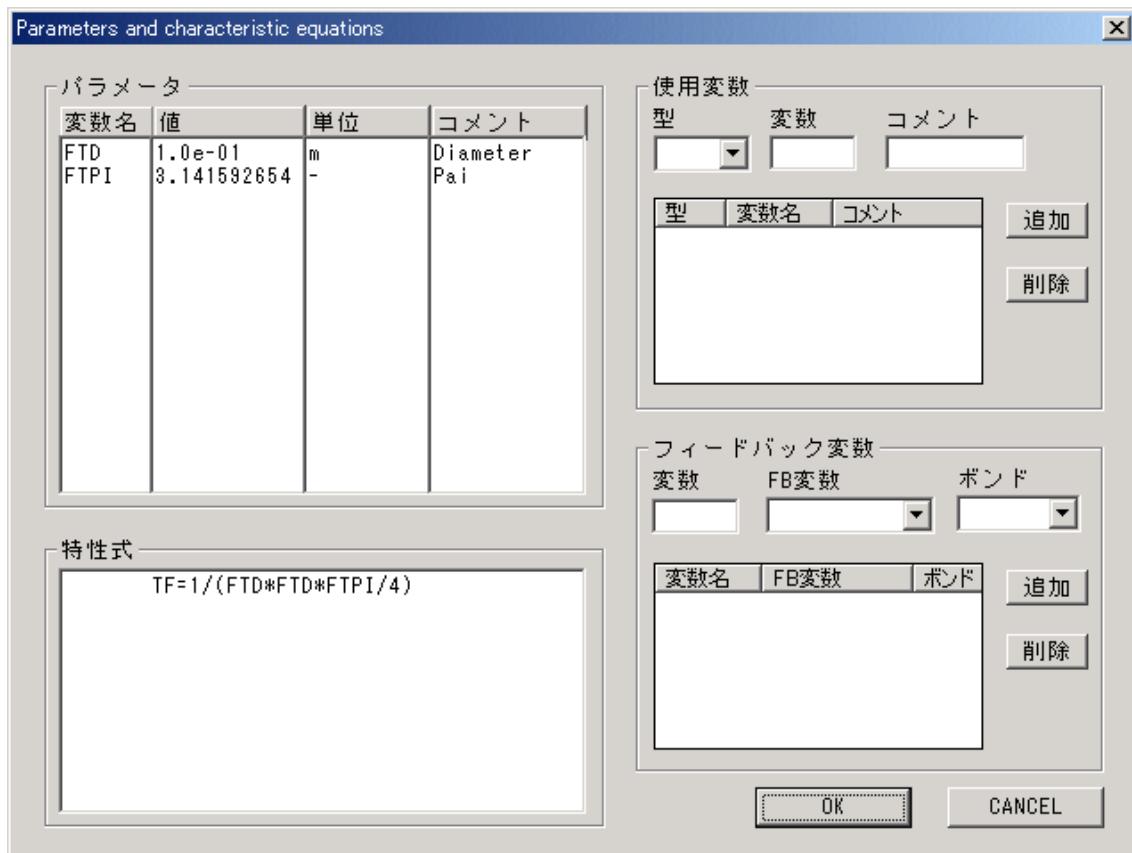
素子 R1



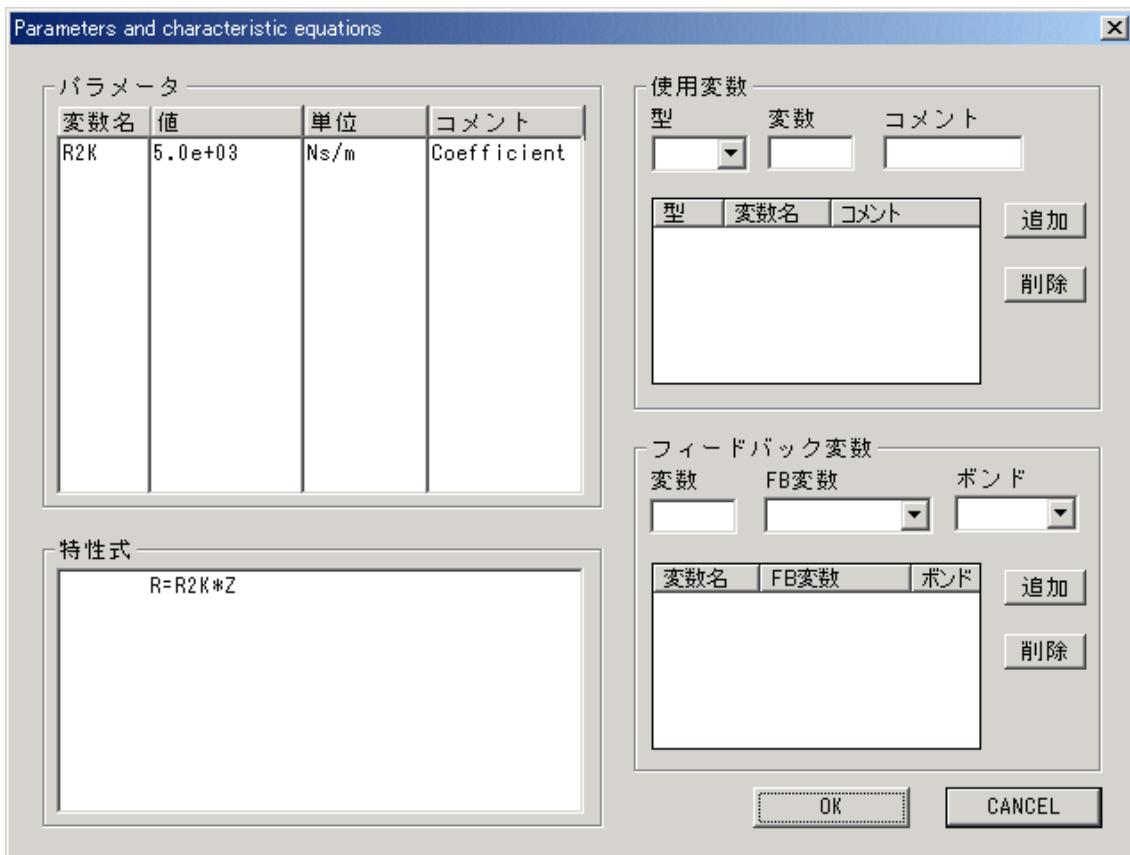
素子 C1



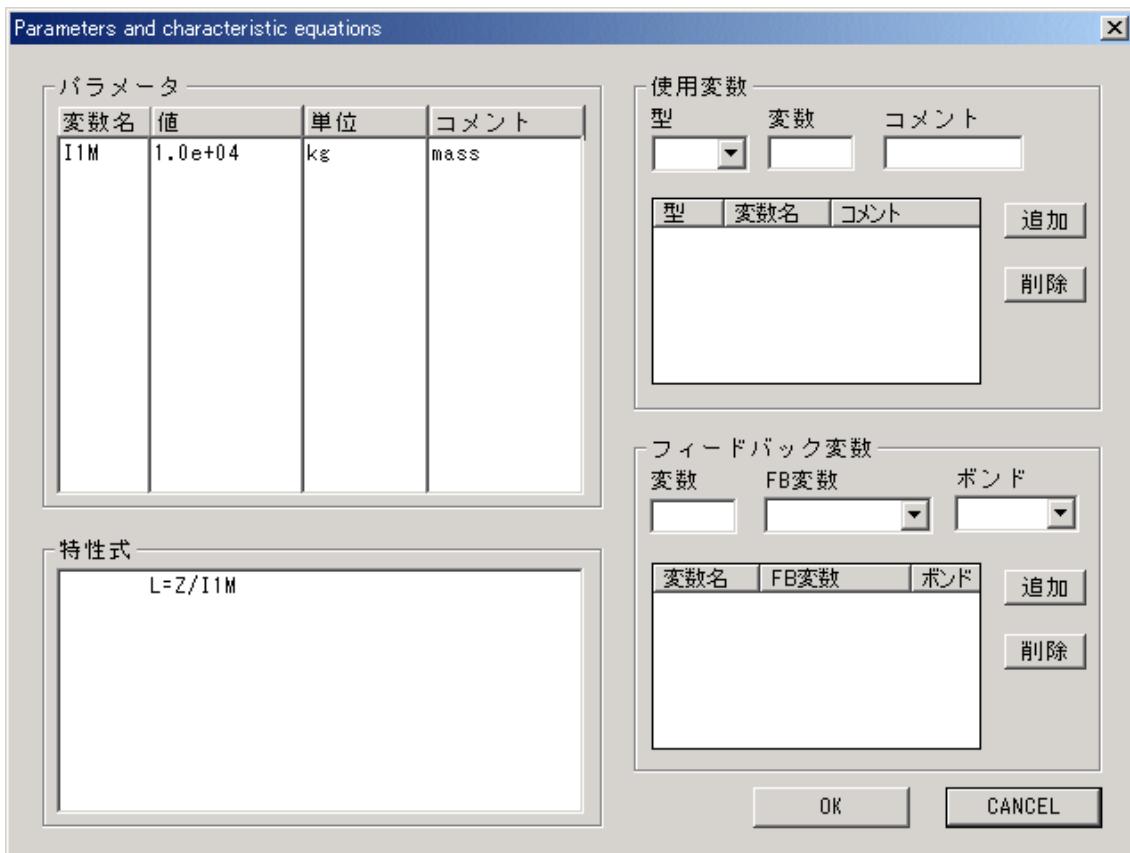
素子 TF1



素子 R2



素子 I1



11-3 計算

図 11-3 次に計算パラメータを以下のように入力します。

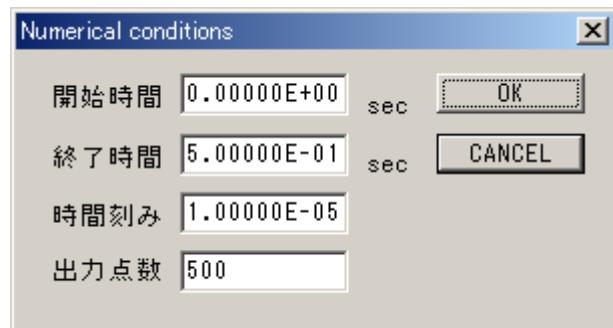


図 11-3 計算パラメータ

図 11-3 のパラメータで計算すると、ボンド番号 4 の EF とボンド番号 7 の DP は図 11-4 のようになります。

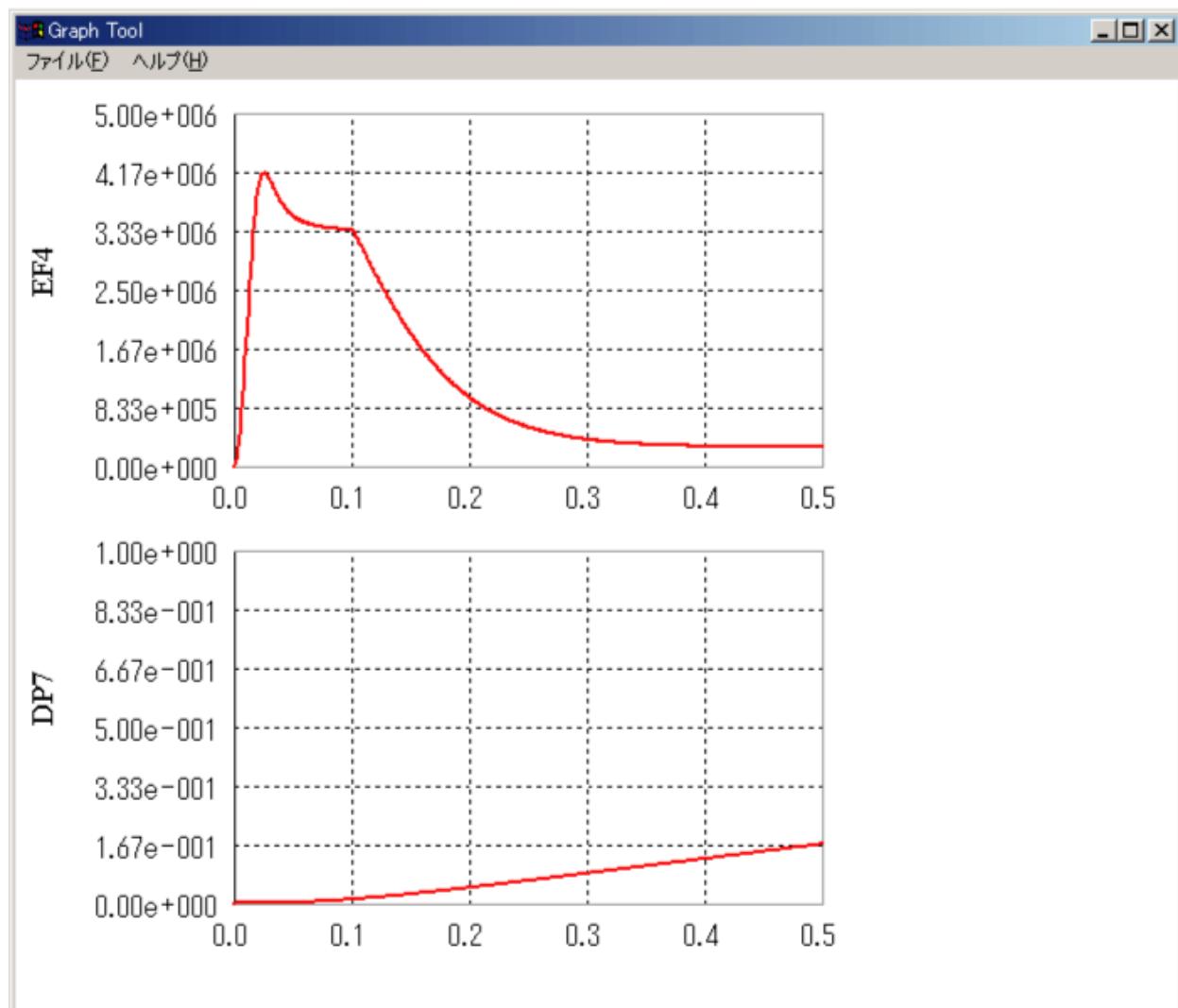


図 11-4 ボンド番号 4 の EF とボンド番号 7 の DP

12. 例 3 電気システムにおけるボンドグラフ

12-1 低域通過フィルター(Low-Pass Filter:LPF)

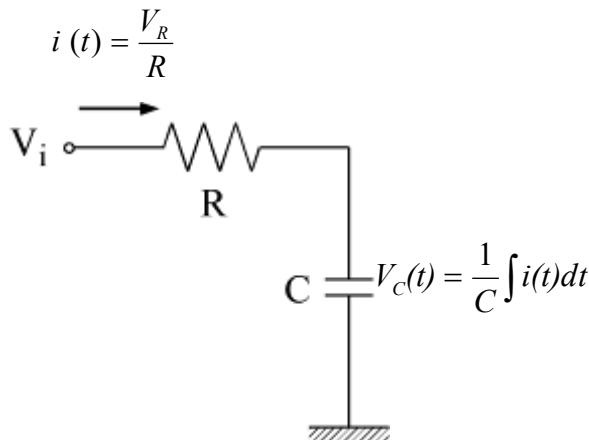


図 12-1 LPF 回路

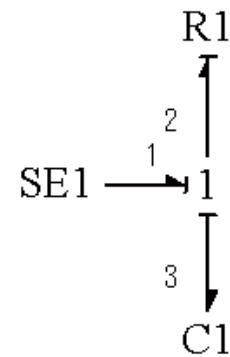


図 12-2 ボンドグラフモデル

図 12-1 は LPF 回路です. V_i は入力電圧, $i(t)$ は電流, V_R , V_C は各素子の端子電圧, R は抵抗, C はコンデンサ, t は時間を表します. 図 11 の物理モデルをボンドグラフにすると図 12-2 のようになります. ここで, SE1 は電源電圧, R1 は抵抗, C1 はコンデンサを表しています.

12-1-1 C version の場合

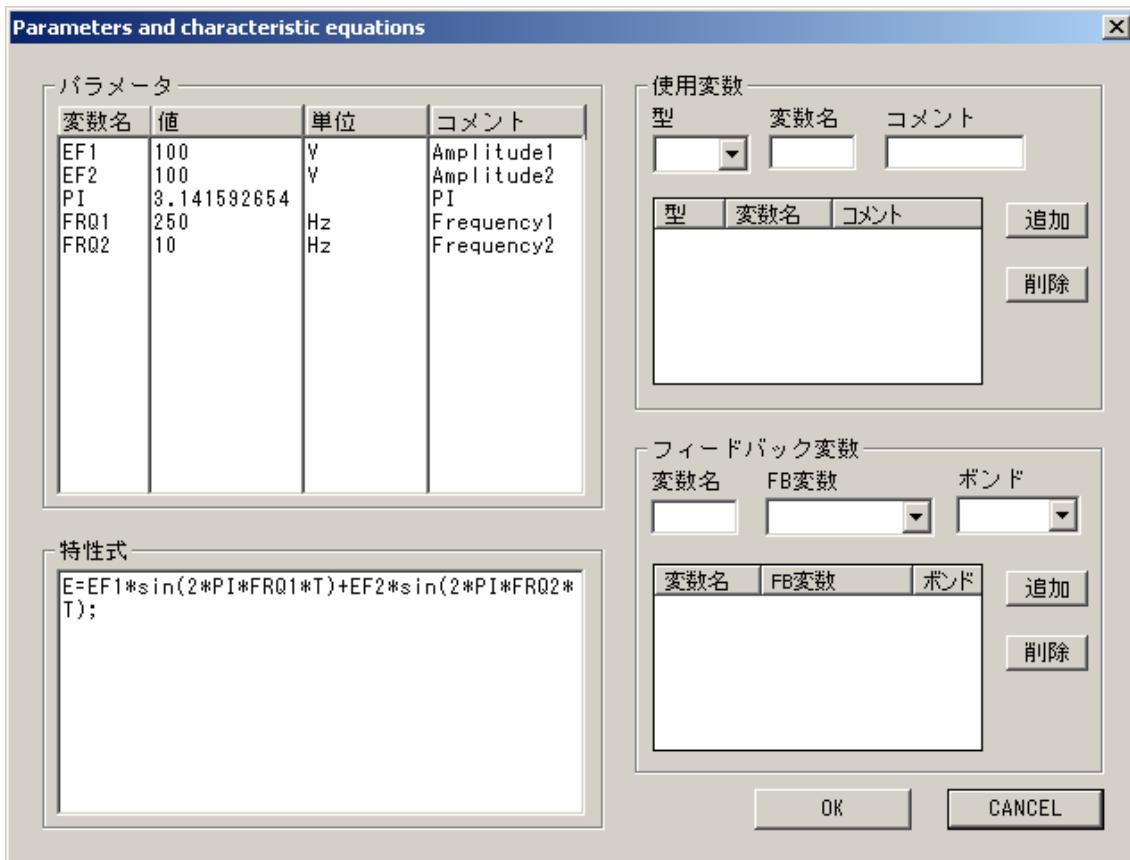
表 12-1 に入力パラメータ情報を示します.

表 12-1 C version における入力パラメータ情報

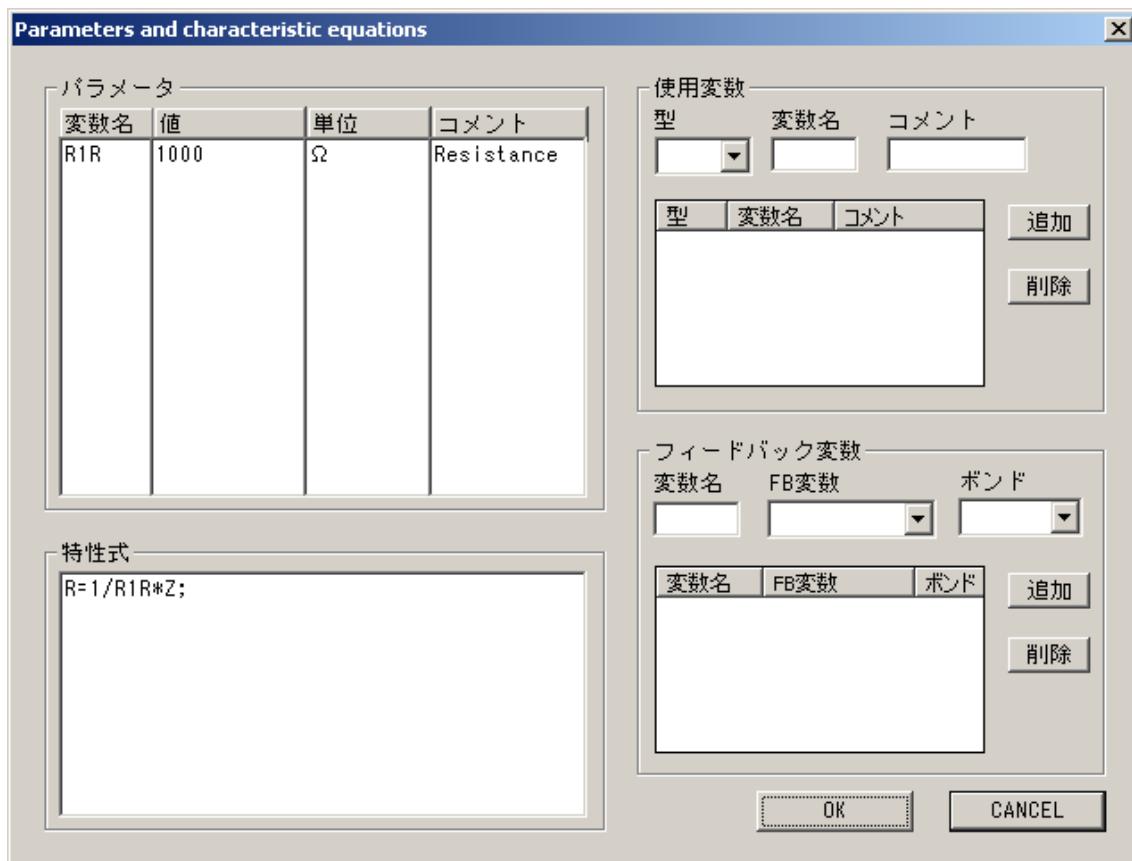
素子名	変数名	値	単位	コメント	特性式
SE1	EF1	100	V	Amplitude1	$E=EF1*\sin(2*PI*FRQ1*T)+EF2*\sin(2*PI*FRQ2*T);$
	EF2	100	V	Amplitude2	
	PI	3.141592654		PI	
	FRQ1	250	Hz	Frequency1	
	FRQ2	10	Hz	Frequency2	
R1	R1R	1000	Ω	Resistance	$R=1/R1R*Z;$
C1	C1C	3.2e-06	F	Capacitance	$L=1/C1C*Z;$

各素子の入力パラメータ情報の入力を示します。

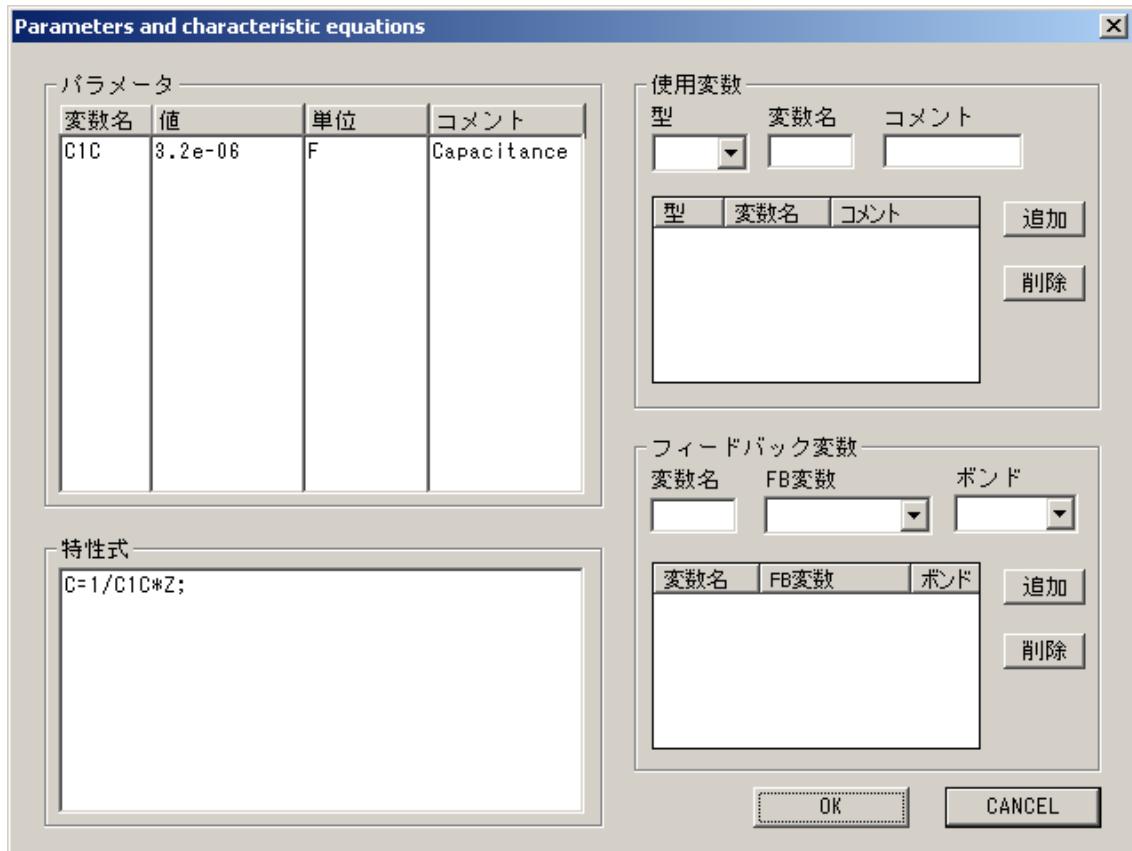
素子 SE1



素子 R1



素子 C1



12-1-2 Fortran Version の場合

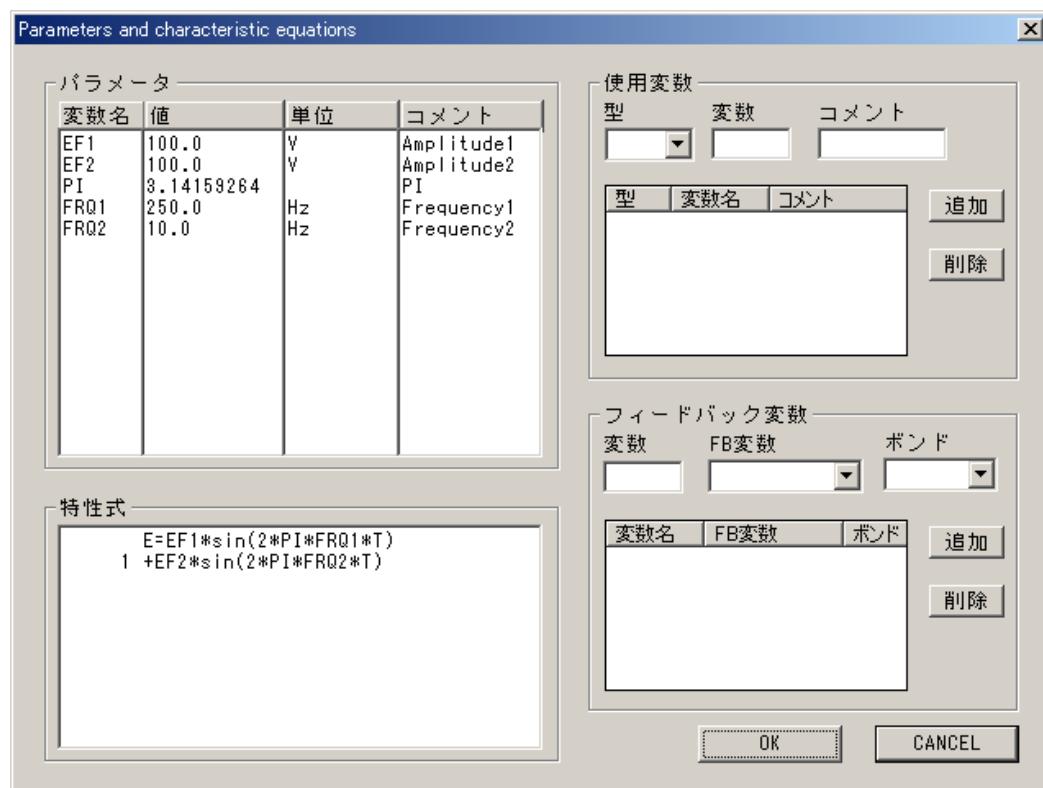
表 12-2 に入力パラメータ情報を示します。

表 12-2 Fortran version における入力パラメータ情報

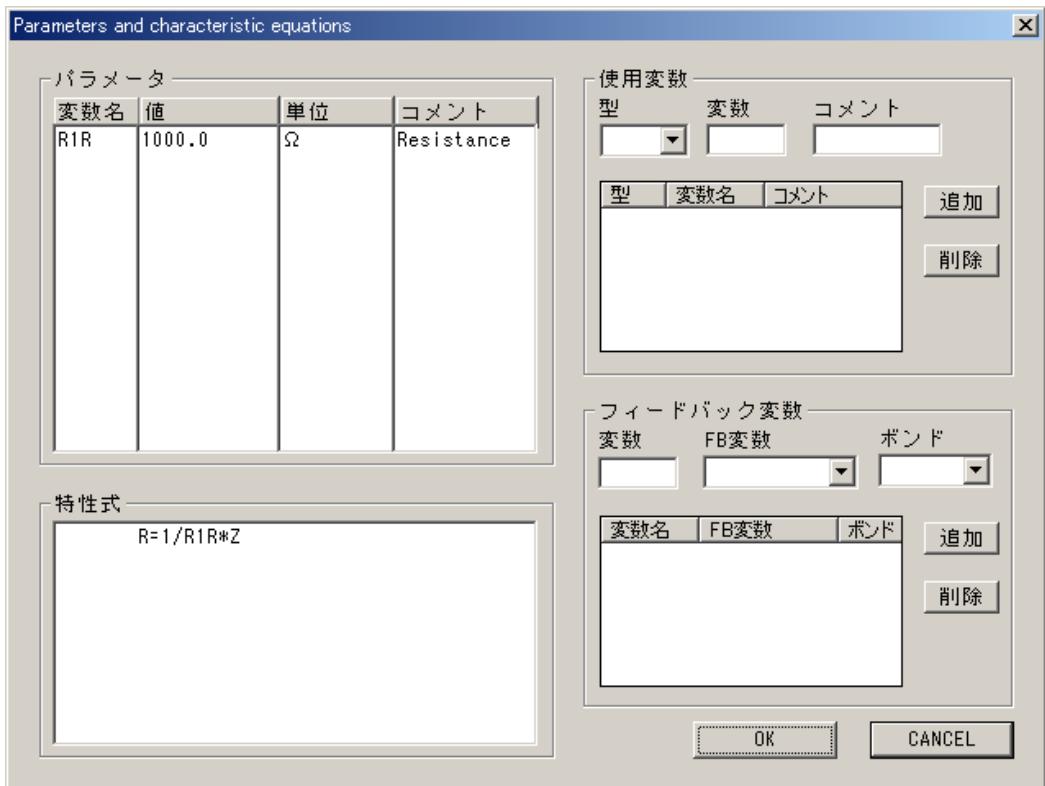
素子名	変数名	値	単位	コメント	特性式
SE1	EF1	100.0	V	Amplitude1	$E=EF1*\sin(2*PI*FRQ1*T)+EF2*\sin(2*PI*FRQ2*T)$
	EF2	100.0	V	Amplitude2	
	PI	3.141592654		PI	
	FRQ1	250.0	Hz	Frequency1	
	FRQ2	10.0	Hz	Frequency2	
R1	R1R	1000.0	Ω	Resistance	$R=1/R1R*Z$
C1	C1C	3.2e-06	F	Capacitance	$L=1/C1C*Z$

各素子の入力パラメータ情報の入力を示します。

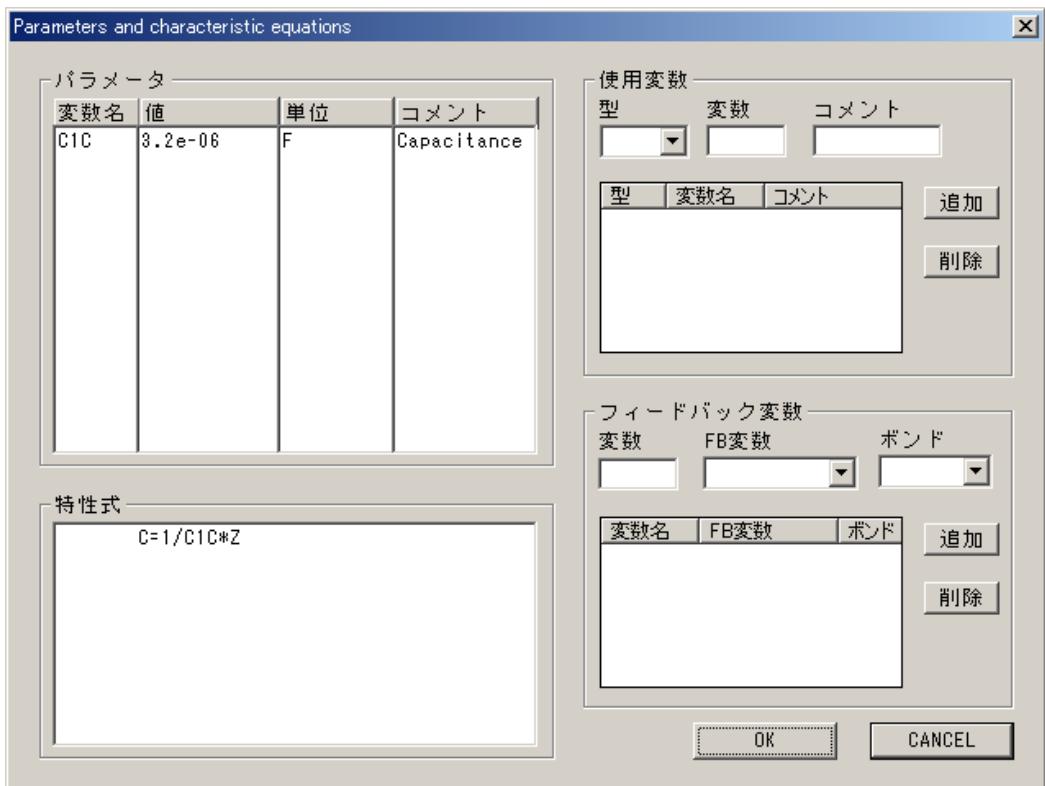
SE1 素子



R1 素子



C1 素子



12-1-3 計算

計算パラメータを図 12-3 のように入力します。

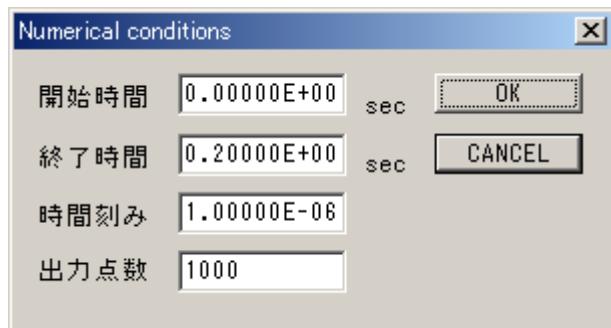


図 12-3 計算パラメータ

図 12-3 のパラメータで計算すると、ボンド番号 1 の EF とボンド番号 3 の EF は図 12-4 のようになります。

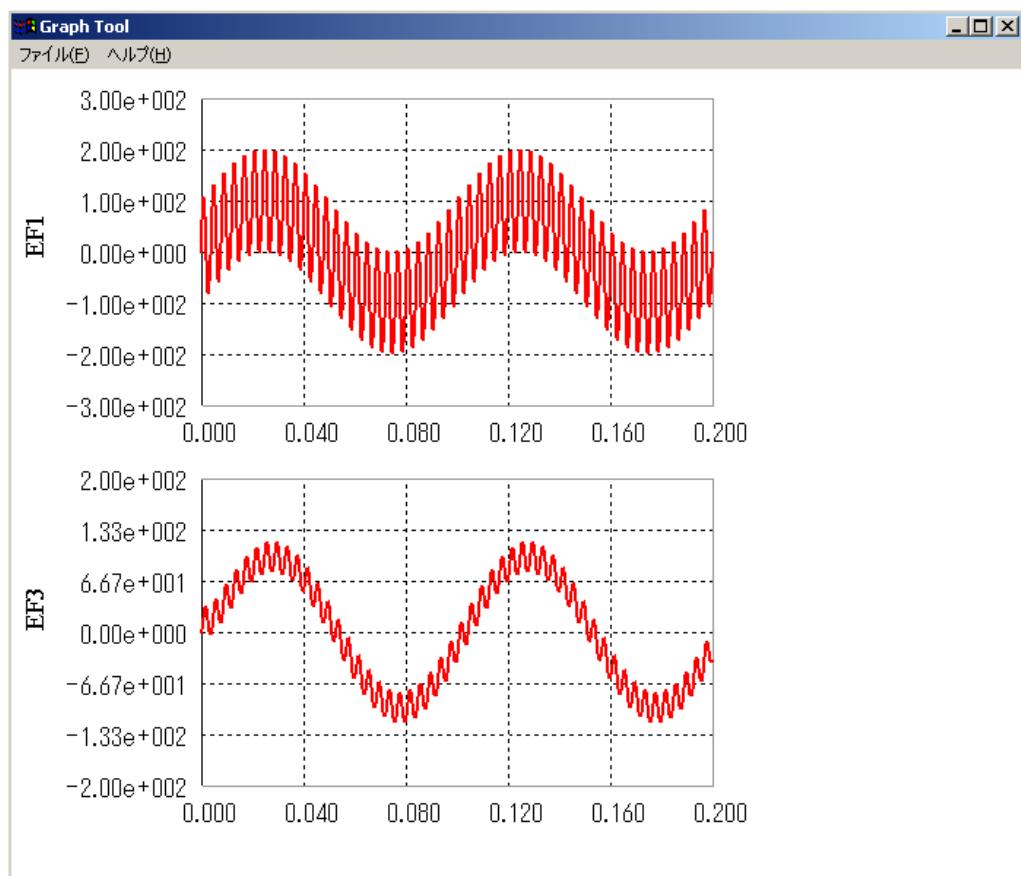


図 12-4 LPF の出力結果

12-2 高域通過フィルター(High-Pass Filter:HPF)

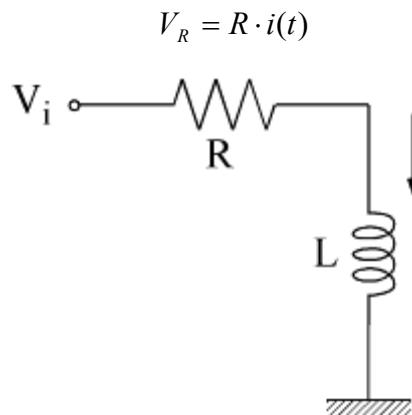


図 12-5 HPF 回路

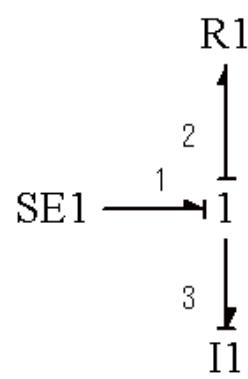


図 12-6 ボンドグラフモデル

図 12-5 は HPF 回路です. V_i は入力電圧, $i(t)$ は電流, V_R , V_L は各素子の端子電圧, R は抵抗, L はコイル, t は時間を表します. 図 12-5 の電気回路をボンドグラフにすると図 12-6 のようになります. ここで, $SE1$ は入力電圧, $R1$ は抵抗, $I1$ はインダクタンスを表しています.

12-2-1 C version の場合

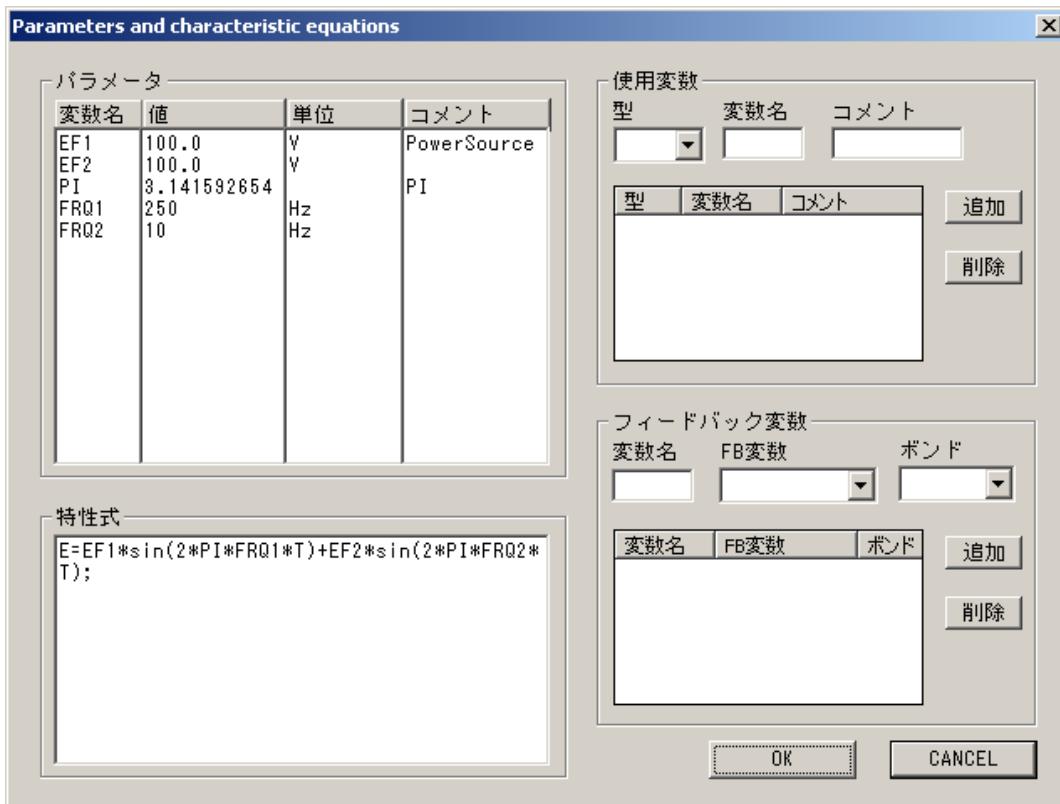
表 12-3 に入力パラメータ情報を示します.

表 12-3 C version における入力パラメータ

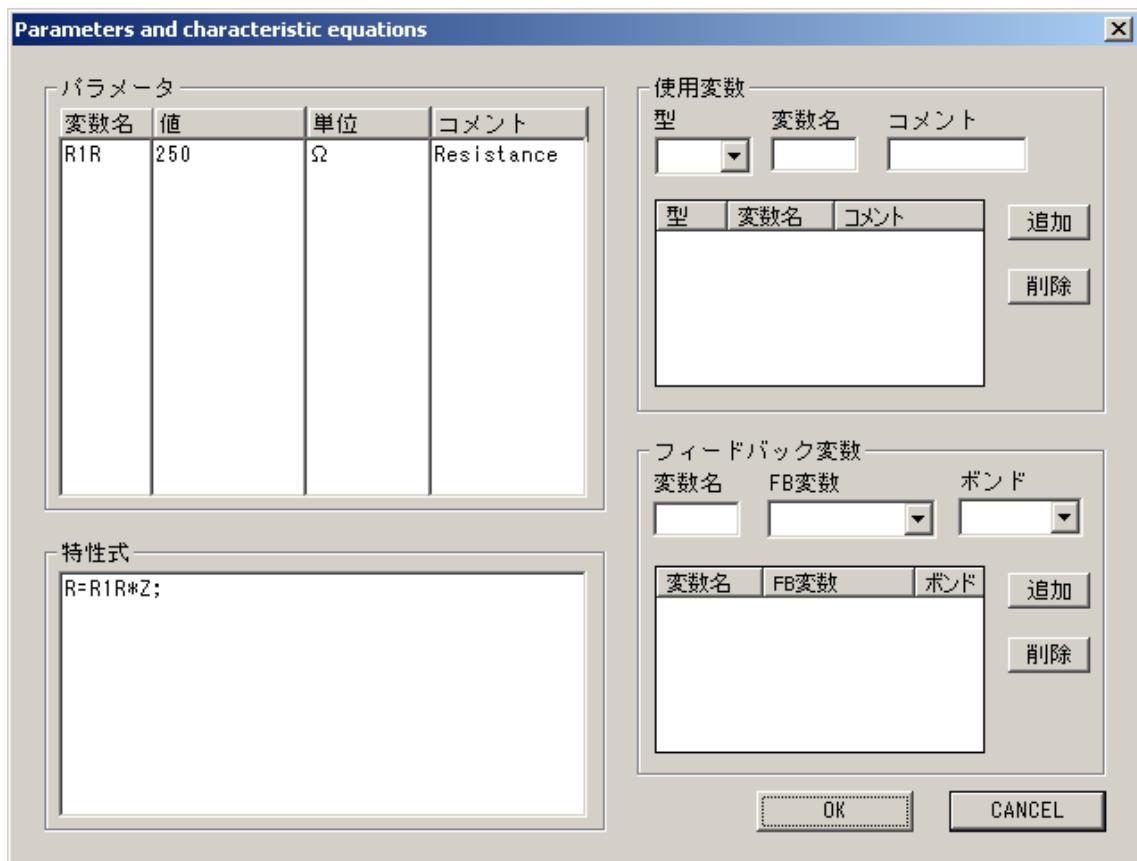
素子名	変数名	値	単位	コメント	特性式
SE1	EF1	100	V	Amplitude1	$E=EF1*\sin(2*PI*FRQ1*T)+EF2*\sin(2*PI*FRQ2*T);$
	EF2	100	V	Amplitude2	
	PI	3.141592654		PI	
	FRQ1	250	Hz	Frequency1	
	FRQ2	10	Hz	Frequency2	
R1	R1R	250	Ω	Resistance	$R=R1R*Z;$
I1	L1L	300e-03	H	Inductance	$L=1/L1L*Z;$

各素子の入力パラメータ情報の入力を示します。

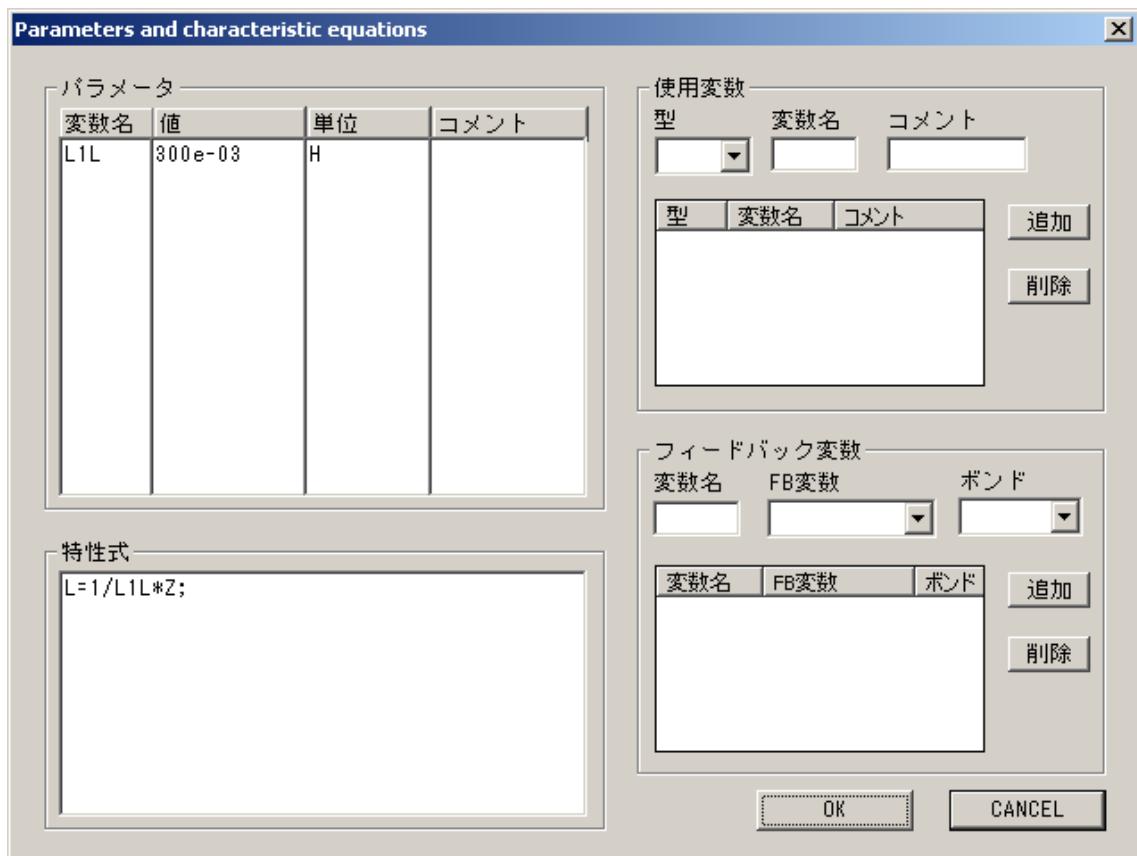
素子 SE1



素子 R1



素子 I1



12-2-2 Fortran Version の場合

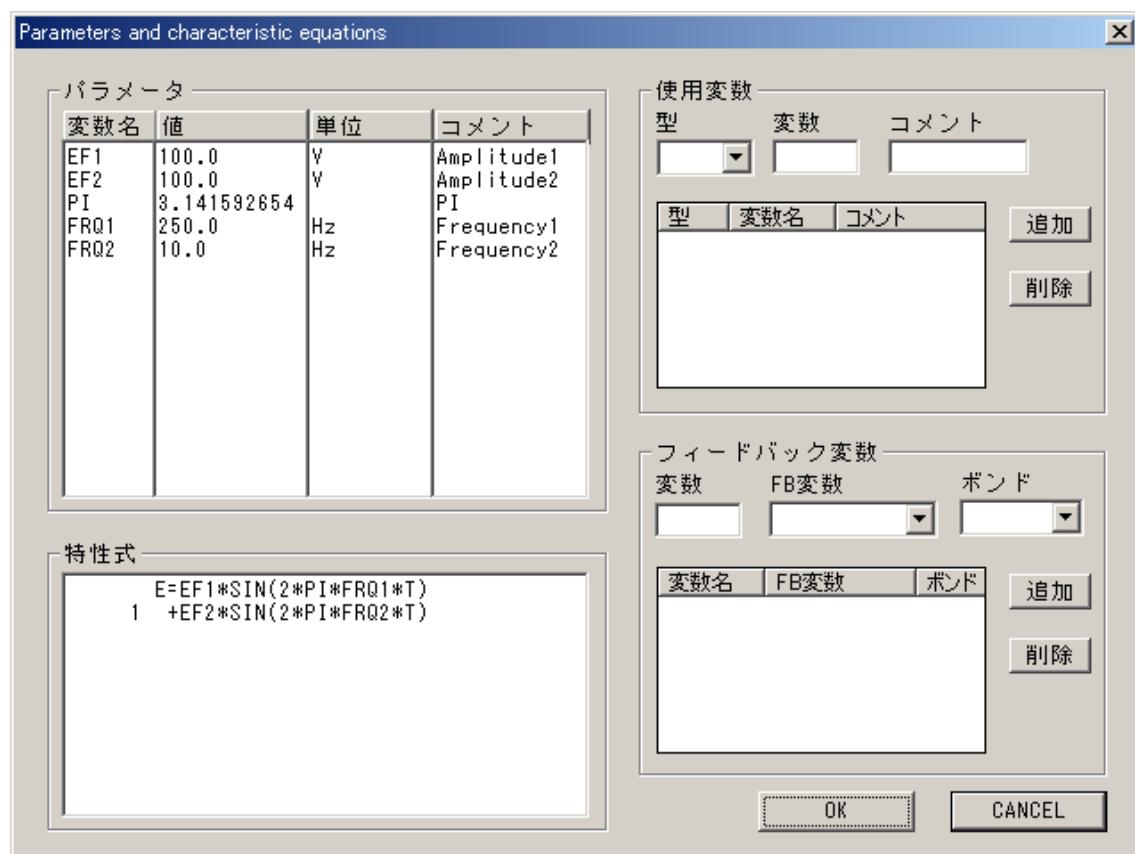
表 12-4 に入力パラメータ情報を示します。

表 12-4 Fortran version における入力パラメータ情報

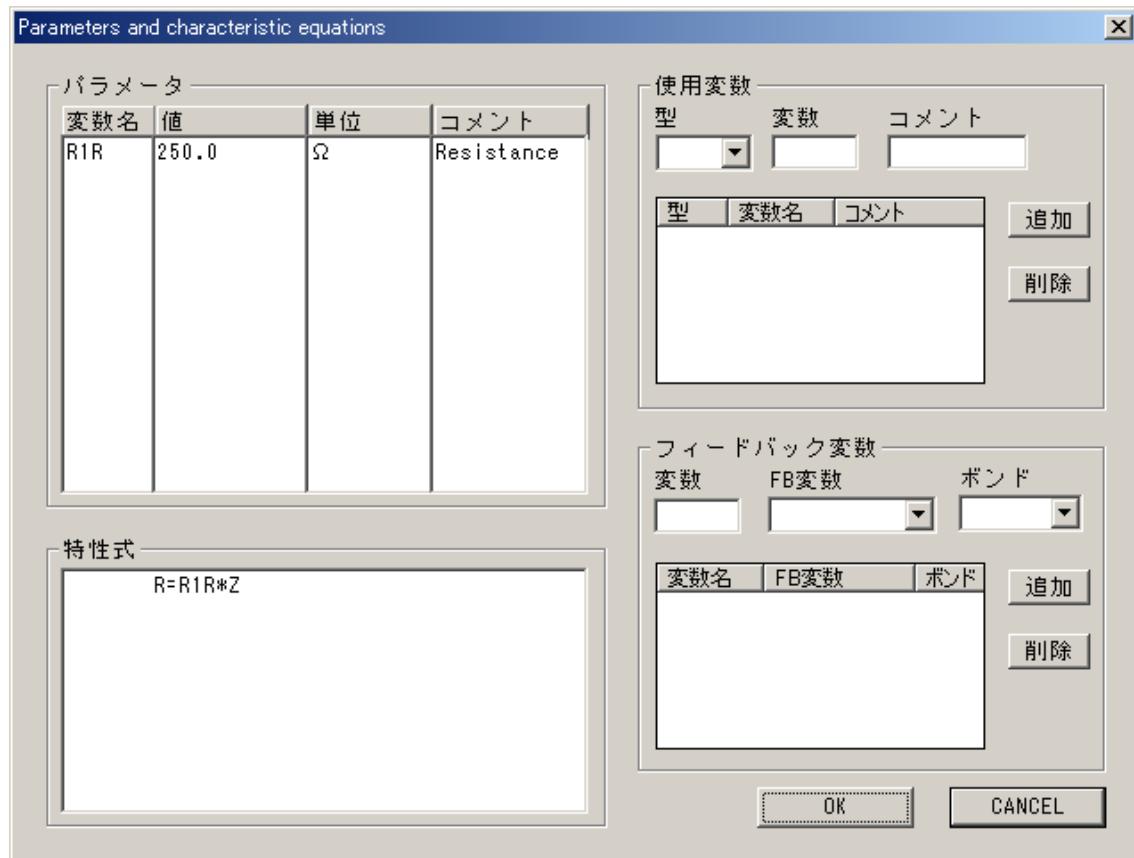
素子名	変数名	値	単位	コメント	特性式
SE1	EF1	100.0	V	Amplitude1	$E=EF1*\sin(2*PI*FRQ1*T)+EF2*\sin(2*PI*FRQ2*T)$
	EF2	100.0	V	Amplitude2	
	PI	3.141592654		PI	
	FRQ1	250.0	Hz	Frequency1	
	FRQ2	10.0	Hz	Frequency2	
R1	R1R	250.0	Ω	Resistance	$R=R1R*Z$
L1	L1L	300.0E-03	H	Inductance	$L=1/L1L*Z$

各素子の入力パラメータ情報の入力を示します。

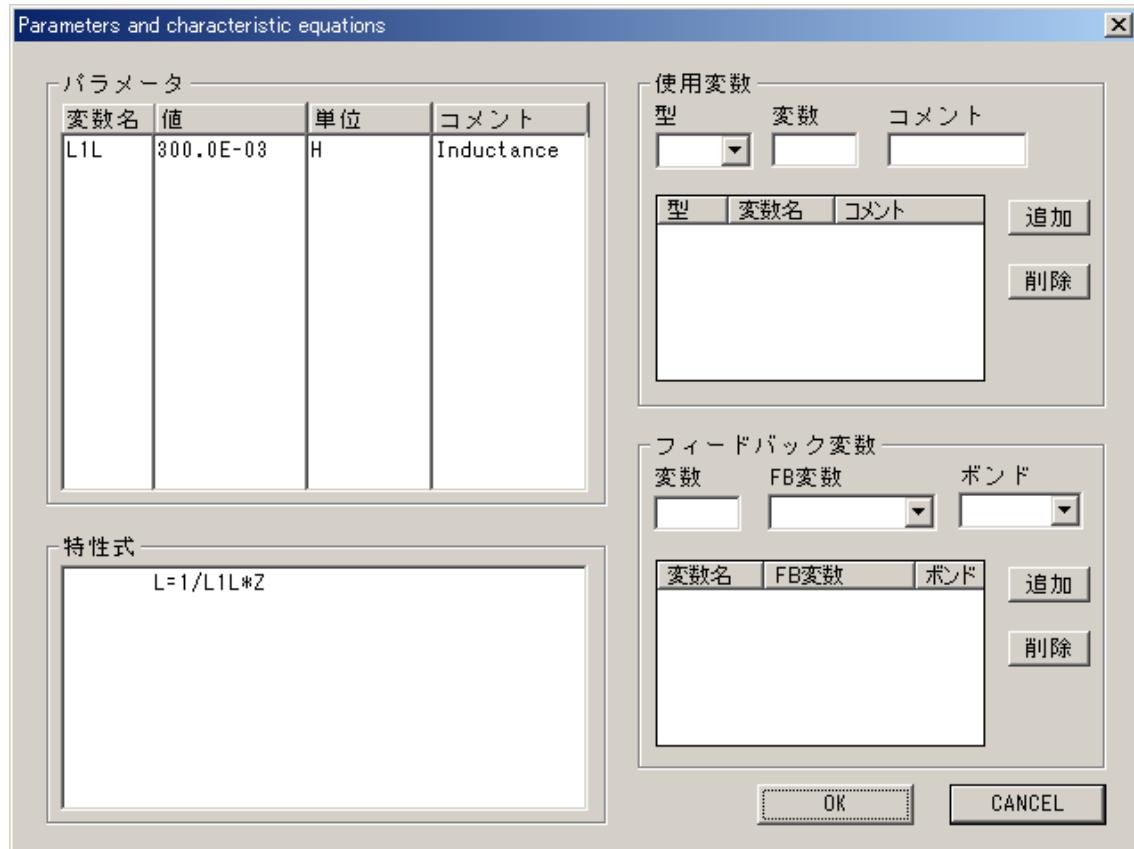
SE1 素子



R1 素子



I1 素子



12-2-3 計算

計算パラメータを図 12-7 のように入力します。

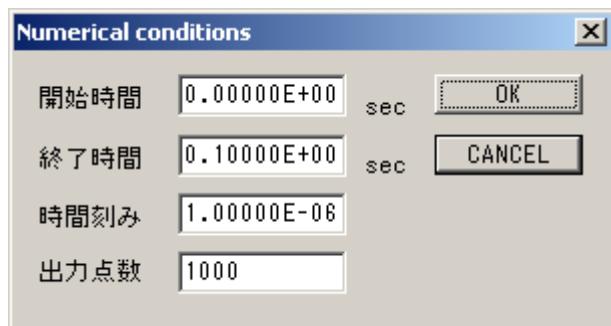


図 12-7 計算パラメータ

図 12-7 のパラメータで計算すると、ボンド番号 1 の EF とボンド番号 3 の EF は図 12-8 のようになります。

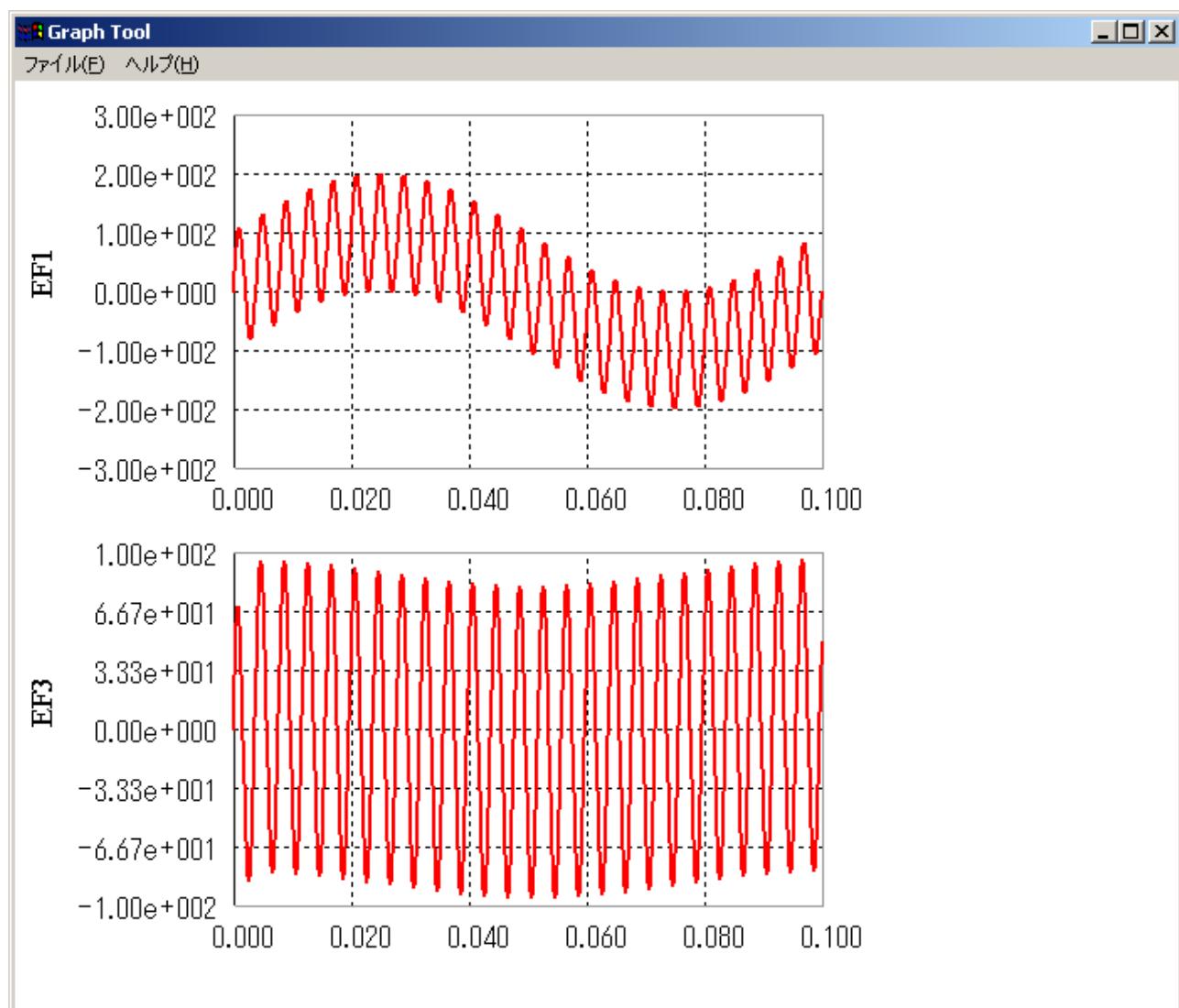


図 12-8 HPF の出力結果

13. BGSPのご利用にあたって

本プログラムを利用する場合、ボンドグラフ法をマスターしていることが前提となります。また、下記に、独自でボンドグラフ法をマスターしたい方のために、参考資料を紹介します。

- (1) Simtec の HP <https://simtec.jp>
ボンドグラフの解説が丁寧になされています。
- (2) Kohda, T., Nakada, T., Kimura, Y. and Mitsuoka, T. : Simulation of Bond Graphs with Nonlinear Elements by Symbolic Manipulation, Bulletin of Mechanical Engineering Laboratory, No. 49, ISSN 0374-2725 (1988)
- (3) 幸田武久,田中和博:パソコンシミュレーションによる油空圧システムの設計, パワーデザイン,25, 8, pp.34~64 (1985)
- (4) 日本機械学会研究協力部会, RC68 油空圧システム省エネルギー調査研究分科会, 成果報告書・I 別冊, BGSP(ボンドグラフ・シミュレーション・プログラム)使用手引書, 日本機械学会(1985)
- (5) (社)日本油空圧学会, ボンド・グラフ・シミュレーション・プログラム(BGSP) の改良に関する研究委員会成果報告書(1992)
- (6) 須田信英, 制御工学,コロナ社(1987)
- (7) J.U.トーマ, 須田信英, ボンドグラフによるシミュレーション, (1996), コロナ社.
- (8) Karnopp, D., Rosenberg, R., System Dynamics: A Unified Approach, (1975), John Wiley & Sons, Inc., New York.
- (9) Rosenberg, R. C., D. C. Karnopp, Introduction to Physical System Dynamics, (1983), McGraw-Hill.
- (10) Bos, A. M., Breedveld, P. C., 1985 Update of the Bond graph Bibliography, Journal of The Franklin Institute, Vol.319, No.1/2(1985), pp269-286, Pergamon Press Ltd.
- (11) Thoma, J. U., Simulation by Bondgraphs, (1990), Springer-Verlag.
- (12) Filippo, J. M., Delgado, M., Brie, C., Painter, H. M., A Survey of Bond Graphs; Theory, Applications and Programs, Journal of The Franklin Institute, Vol.328, No.5/6(1991), pp565-605, Pergamon Press Ltd.
- (13) Thoma, J. U., Ould Bouamama, B., Modeling and Simulation in Thermal and Chemical Engineering. A Bond Graph Approach, (2000), Springer Verlag, Berlin.
- (14) W. Borutzky, Bond Graphs Methodology, (2010), Springer Link.
- (15) 田中和博, 設計のための1DCAE 概念と実現技術, 1DCAE レクチャーシリーズ, (2020), 日本機械学会, pp82-115.

付録. 1 Fortran と C++ の比較

・Fortran と C++ の比較

以下に基本的な処理について、Fortran と C++ の違いを示します。

また最後に Fortran, C++ による例文をのせ、構文の違いを示します。

● 加算($w=x+y$)

Fortran $W=X+Y$
C++ $w=x+y;$

● 減算($w=x-y$)

Fortran $W=X-Y$
C++ $w=x-y;$

● 乗算($w=x \times y$)

Fortran $W=X*Y$
C++ $w=x*y;$

● 除算($w=x \div y$)

Fortran $W=X/Y$
C++ $w=x/y;$

● 剰余($x \div y=z \cdots w$)

Fortran $W=MOD(X,Y)$
C++ $w=x \% y;$

● べき乗($w=x^y$)

Fortran $W=X^{**}Y;$
C++ $w=pow(x,y);$

● 平方根($w=\sqrt{X}$)

Fortran $W=SQRT(X)$
C++ $w=sqrt(x);$

● 指数関数($w=e^x$)

Fortran $W=EXP(X)$
C++ $w=exp(x);$

● 自然対数関数($w=\ln(x)$)

Fortran $W=LOG(X)$
C++ $w=log(x);$

● 常用対数関数($w=\log(x)$)

Fortran $W=LOG10(X)$
C++ $w=log10(x);$

● 正弦関数($w=\sin(x)$)

Fortran $W=SIN(X)$
C++ $w=sin(x);$

● 余弦関数($w=\cos(x)$)

Fortran $W=COS(X)$
C++ $w=cos(x);$

● 正接関数($w=\tan(x)$)

Fortran $W=TAN(X)$
C++ $w=tan(x);$

● 逆正弦関数($w=\arcsin(x)$)

Fortran $W=ASIN(X)$
C++ $w=asin(x);$

● 逆余弦関数($w=\arccos(x)$)

Fortran $W=ACOS(X)$
C++ $w=acos(x);$

● 逆正接関数($w=\arctan(x)$)

Fortran $W=ATAN(X)$
C++ $w=atan(x);$

● 双曲線正弦関数

Fortran $W=SINH(X)$
C++ $w=sinh(x);$

● 双曲線余弦関数

Fortran $W=COSH(X)$
C++ $w=cosh(x);$

● 双曲線正接関数

Fortran $W=TANH(X)$
C++ $w=tanh(x);$

● 絶対値(引数が実数型の場合)

Fortran $W=ABS(X)$
C++ $w=fabs(x);$

● ループ(0 から 10 までの足し算)

Fortran DO 30 I=0,10
 J=J+I
30 CONTINUE
C++ for(i=0;i<=10;i++) j=j+i;

●条件文 1(もし $a > b$ ならば…)

```
Fortran  IF(A.GT.B) THEN
          c ここに処理を書く
          END IF
C++      if(a>b)
{
    //ここに処理を書く
}
```

●条件文 2(もし $a < b$ ならば…)

```
Fortran  IF(A.LT.B) THEN
          c ここに処理を書く
          END IF
C++      if(a>b)
{
    //ここに処理を書く
}
```

●条件文 3(もし $a \geq b$ ならば…)

```
Fortran  IF(A.GE.B) THEN
          c ここに処理を書く
          END IF
C++      if(a>=b)
{
    //ここに処理を書く
}
```

●条件文 4(もし $a = b$ ならば…)

```
Fortran  IF(A.EQ.B) THEN
          c ここに処理を書く
          END IF
C++      if(a==b)
{
    //ここに処理を書く
}
```

●入力(変数 x を入力)

```
Fortran  READ(5,*) X
C++      scanf("%f",&x);
```

●出力(変数 x を出力)

```
Fortran  WRITE(6,*) 'X=',X
C++      printf("x=%f",x);
```

●データ型

整数

Fortran INTEGER

C++ int

実数

Fortran REAL

C++ float

倍精度実数

Fortran DOUBLE PRECISION

C++ double

文字型

Fortran CHARACTER

C++ char

●例文(シンプソンの積分)

シンプソンの積分公式をもちい, $Y = \sqrt{4 - x^2}$ の $x=0$ から $x=2$ までの積分値を求める.

変数の説明

A,B 積分区間
M 積分範囲の分割数
H 区間幅
FUNC,T 被積分関数とその変数
SUMO,SUME 値を一時的に入れる変数
S 積分値を代入する変数

○Fortran の場合

```
C      シンプソンの公式を用いた積分
C      変数の宣言
      INTEGER M
      REAL FUNC,A,B,H,S,T,SUMO,SUME

C      関数の定義
      FUNC(T)=SQRT(4-T**2)

C      変数の読み込み
      READ(5,100) M,A,B
100   FORMAT(I8,2F10.0)

C      区間幅の決定
      H=(B-A)/M
      SUMO=0
      SUME=0

C      積分計算
      DO 10 I=1,M/2-1
          SUMO=SUMO+FUNC(A+H*(2*I-1))
          SUME=SUME+FUNC(A+H*(2*I))
10     CONTINUE
      SUMO=SUMO+FUNC(B-H)
      S=(FUNC(A)+FUNC(B)+4*SUMO+2*SUME)*H/3

C      出力
      WRITE(6,200) 'M=',M,' H=',H,' S=',S
200   FORMAT(' ',A8,I8,A8,F10.7,A8,F10.7)
      END

C      Fortran における実行結果
C      100,0,2
C      M= 100    H= 0.0200000    S= 3.1411333
```

○C++の場合

```
#include<stdio.h>
#include<math.h>

//Class の定義
class variable{
public:
    float s;
    float h;
    float func(float t){
        return (float)(sqrt(4-pow(t,2))); //関数の定義
    }
} simpson;

int main(void){
    //変数の定義
    int i;
    float sumo,sume,m,a,b;

    //入力
    scanf("%f %f %f",&m,&a,&b);

    //区間幅 simpson.h の決定
    simpson.h=(b-a)/m;
    sumo=0.0;
    sume=0.0;

    //積分値の計算
    for(i=1;i<=m/2-1;i++){
        sumo=sumo+simpson.func(a+simpson.h*(2*i-1));
        sume=sume+simpson.func(a+simpson.h*(2*i));
    }

    sumo=sumo+simpson.func(b-simpson.h);
    simpson.s=(simpson.func(a)+simpson.func(b)
        +4*sumo+2*sume)*simpson.h/3;

    //出力
    printf("M=%4.0f H=%f S=%f\n",m,simpson.h,simpson.s);
    return 0;
}

//c++における実行結果
//100 0 2
//M= 100 H= 0.0200000 S= 3.1411333
```

付録. 2 ボンドグラフ(BG)に関する基礎事項

1. 因果律とストローク



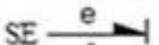
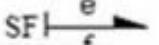
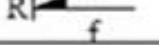
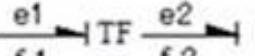
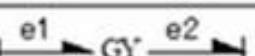
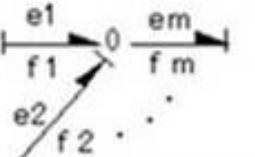
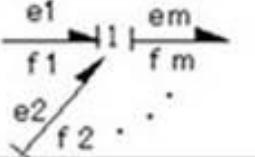
2. エネルギーフローの観点から物理系を構成する要素と BG 素子

	エネルギー・フローの観点から物理系を構成する要素	各要素に対応しているボンドグラフ素子
1	エネルギー供給要素	SE-素子 (Source of Effort), SF-素子 (Source or Flow)
2	エネルギー貯蔵要素	C-素子 (Capacitance), I-素子 (Inertance)
3	エネルギー消費要素	R-素子 (Resistance)
4	エネルギー伝達要素	TF-素子 (Transformer), GY-素子 (Gyrator)
5	エネルギーの分歧・合流を表す加算点	0-接点と1-接点の2種類の接点

3. 物理的なシステム構成要素に対応する BG 要素

素子	機械系	流体系	電気系
ボンド	シャフト, ロッド	管路	電線
SE-素子	外力	圧力制御付ポンプ	電圧源
SF-素子	固体壁, 伝動軸	容積形ポンプ	電流源
I-素子	質量	流体慣性	コイル
	フライホイール	流体質量	インダクタンス
	剛体の慣性モーメント		
C-素子	ばね	アクチュエータ	コンデンサ
	弾性変形	流体の圧縮性	
	ねじれ	タンク, 容積効果	
R-素子	摩擦, 摩擦トルク, トルク損失	流体抵抗, 流体摩擦	電気抵抗
	すべり(減速), ダンパー	紋り, 弁, 漏れ	
TF-素子	てこ	ピストン(受圧面)	変圧器
	径違い歯車	シリンド	
GY-素子	ジャイロ	ターボ形流体機械	モータ
0-接点	遊星歯車または復元てこによる結合	配管の分歧・合流部	並列回路
1-接点	固定継ぎ手による結合, 移動物体	直列, 移動流体	直列回路

4. BG の要素表現と特性関数

基本素子	記号	特性関数
SE		$e = E(t)$
SF		$f = F(t)$
R		$e = R(f)$
		$f = R^{-1}(e)$
C		$e = C \left(\int f dt \right)$
I		$f = L \left(\int e dt \right)$
TF (n:比例 係数)		$e2 = e1/(n)$ $f1 = f2/(n)$
		$e1 = (n)e2$ $f2 = (n)f1$
GY (m:比例 係数)		$f2 = e1/(m)$ $f1 = e2/(m)$
		$e1 = (m)f2$ $e2 = (m)f1$
0-接点		$e_i = \text{common}$ $\sum_{i=1}^n f_i = 0$
1-接点		$f_i = \text{common}$ $\sum_{i=1}^n e_i = 0$

$$v = \frac{1}{M} \int \left\{ F_{input} - \left(Dv + K \int v dt \right) \right\} dt + v(0)$$

一方, C 素子の特性則は,

$$F = Kx = K \int v dt + F(0)$$

(3)上記の運動方程式は, サブシステムの動特性を表現するには三要素(慣性=I 素子, ばね=C 素子, 抵抗=R 素子)とそのサブシステムへの直接入力, あるいは伝達要素からの入力から成り立っていることを表している.

積分因果律を採用する際の留意点

(1)左表には微分記号はどこにも見られない. これは動特性を表現する連立方程式を時系列で数値計算する際に数値積分を用いるため, 積分因果律を採用している為である.

(2)システム(質点系)の運動方程式

$$M \frac{d^2 x}{dt^2} + D \frac{dx}{dt} + Kx = F_{input}$$

左辺第 1 項は慣性力, 第 2 項は抵抗力, 第 3 項はばね力, 右辺に外力(入力).

パワー保存則に基づくボンドグラフでは, 力 F と速度 v が基本変数となる.

$$M \frac{dv}{dt} + Dv + K \int v dt = F_{input}$$

これに積分因果律を適用して速度を求めようとすると,

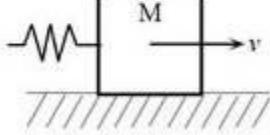
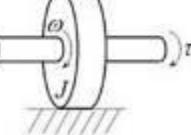
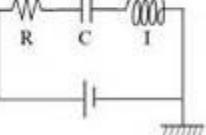
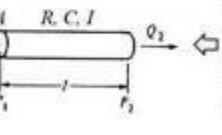
という形になる. これが I 素子の特性則である. ニュートンの運動方程式の積分形表現である.

という形になる. これは, ばね効果を考えると理解しやすい.

5. 0接点と1接点の物理的意味

	0接点(common effort junction)	1接点(common flow junction)
電気系	キルヒ霍ッフの電流則（電気回路網中の接続点における流入する電流の総和は流出する電流の総和に等しい），電圧が共通	キルヒ霍ッフの電圧則（電気回路網の閉ループの周りの電圧上昇の代数和は電圧降下の代数和に等しい），電流が共通
機械系	等しい力が作用していて各要素の速度の総和が0となるような状況に対する幾何学的適合性，作用している力が共通	一つの速度に関連する力の平衡関係（たとえば，慣性項が含まれる場合であれば質量に関するニュートンの法則），速度が共通
流体系	配管の結合点における流量の保存則（流出流量は流入流量に等しい），圧力が共通	流体回路網の閉ループの周りの圧力降下の総和は0でなければならない，流量が共通

6. 機械系と電気系の BG 要素の特性式(式の形式は同じで係数のみ異なる)

素子	表記法	動力(パワー) $effort \times flow = F \times V = \tau \times \omega = e \times i = P \times Q$			
		機械直線系	機械回転系	電気系	流体系
R	R 	ダンバ等減衰力 $F = R \cdot v$	摩擦トルク $\tau = R \cdot \omega$	オームの法則 $e = R \cdot i$	流体抵抗 $P_3 = R \cdot Q_3$
	R 	$v = (1/R) \cdot F$	$\omega = (1/R) \cdot \tau$	$i = (1/R) \cdot e$	$Q_3 = (1/R) \cdot P_3$
C	C 	フックの法則 $F = k \int v dt$	ねじれ剛性 $\tau = \frac{1}{G} \int \omega dt$	コンデンサの電流 $i = C \cdot \frac{de}{dt}$ $e = \frac{1}{C} \int i dt$	容積効果 $Q(t) = C_3 \frac{d\Delta P(t)}{dt}$ $P_3 = \frac{1}{C_3} \int Q_3 dt$
	I 	ニュートンの法則 $F = M \cdot \frac{dv}{dt}$	はづみ車 $\tau = J \cdot \frac{d\omega}{dt}$	コイルの電位差 $e = L \cdot \frac{di}{dt}$ $i = \frac{1}{L} \int e dt$	流体慣性 $P_3 = \frac{\rho l}{A} \cdot \frac{dQ_3}{dt}$ $Q_3 = \frac{A}{\rho l} \int P_3 dt$
管路 $C_3 = V_0 / K$ 断面一定の大気開放タンク $C_3 = \frac{A}{\rho g}$ ガス入りアキュムレータ $C_3 = \frac{V_0}{\gamma P_0}$					
フルモデル (R, C, I要素を使用) に相当する各系での基本要素					
					
					

付録. 3 ボンドグラフシミュレーションツールの考え方

ボンドグラフシミュレーションツールの考え方

3.1 解析の流れ

ボンドグラフシミュレーションツールを用いた動特性解析の流れを説明する。

1. ボンドグラフを作成する
2. ボンドグラフを基に、システムの支配方程式の作成する
3. 作成した支配方程式を C 言語、または FORTRAN の関数に変換する
4. 変換された支配方程式を Runge-Kutta 法を用いて解くことにより動特性を求める

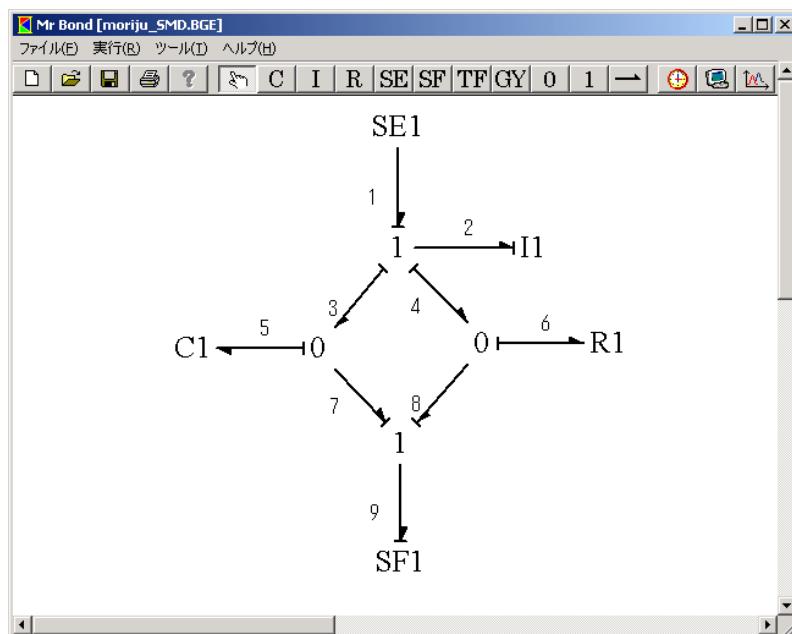


Fig 3.1 Bondgraph simulation tool

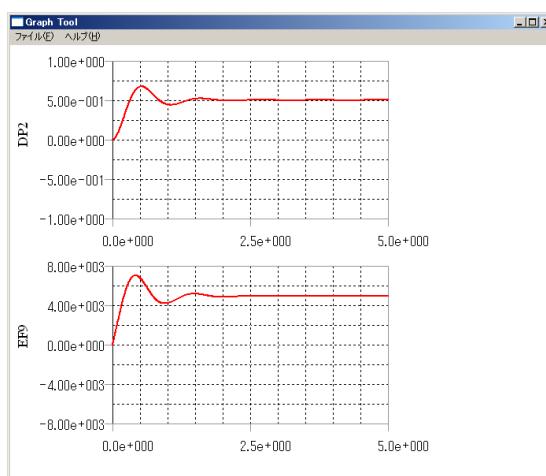


Fig 3.2 System dynamic characteristics

3.2 支配方程式を表示するまでの流れ

3.2.1 支配方程式の作成

支配方程式を作成するまでの手順を図3.1のボンドグラフを例にとり説明する。それを説明する過程ではボンドグラフシミュレーションツールの内部での形と一般的な式の形の両方を示す。ボンドグラフシミュレーションツールの内部において支配方程式は(3.1)式のような形で格納されている。(3.1)式を一般的な式の形にすると(3.2)式になる。

$$\begin{aligned} \text{DX[0]} &= (-C1(X[1]) + E1() - R1(0, (-F1() + L1(X[0])))); \\ \text{DX[1]} &= (L1(X[0]) - F1()); \end{aligned} \quad (3.1)$$

$$\begin{aligned} \frac{dx_0}{dt} &= (-C_1(x_1) + E_1(t) - R_1(-F_1(t) + L_1(x_0))) \\ \frac{dx_1}{dt} &= (L_1(x_0) - F_1(t)) \end{aligned} \quad (3.2)$$

ただし、 C_1 は(2.2)式における $\int f(t)dt$ の係数、 L_1 は(2.3)式における $\int e(t)dt$ の係数、 R_1 は(2.7)式における $f(t)$ の係数である。 $E_1(t)$ は入力エフオートの関数、 $F_1(t)$ は入力フローの関数を表す。ここで、図3.1におけるボンド番号及びパワー変数と x_0 及び x_1 の関係を以下に示す。

$$\begin{aligned} x_0 &= \int e_2 dt \\ x_1 &= \int f_5 dt \end{aligned} \quad (3.3)$$

(3.3)式を(3.1)式に代入すると(3.4)式になる。(3.4)式を一般的な式の形にすると(3.5)式になる。(3.4)式中の\$はプログラム内において積分記号を表すために用いる文字である。

$$\begin{aligned} e2 &= (-C1\$f5d + E1t - R1(-F1t + L1\$e2d)) \\ f5 &= (L1\$e2d - F1t) \end{aligned} \quad (3.4)$$

$$\begin{aligned} e_2 &= \left(-C \int f_5 dt + E_1(t) - R_1 \left(-F_1(t) + L_1 \int e_2 dt \right) \right) \\ f_5 &= \left(L_1 \int e_2 dt - F_1(t) \right) \end{aligned} \quad (3.5)$$

括弧がついた状態では変形しづらいので、(3.4)式を展開すると(3.6)式になる。また、(3.6)式を一般的な形にすると(3.7)式になる。

$$\begin{aligned} e2 &= -C1\$f5d + E1t + R1F1t - R1L1\$e2d \\ f5 &= L1\$e2d - F1t \end{aligned} \quad (3.6)$$

$$\begin{aligned} e_2 &= -C \int f_5 dt + E_1(t) + R_1 F_1(t) - R_1 L_1 \int e_2 dt \\ f_5 &= L_1 \int e_2 dt - F_1(t) \end{aligned} \quad (3.7)$$

ここで、C 素子、I 素子の特性式((2.3)式、(2.4)式)より 2 番ボンド及び 5 番ボンドにおけるエフオートとフローの関係を以下に示す。

$$\begin{aligned} e_2 &= \frac{1}{L_1} \dot{f}_2 \\ f_5 &= \frac{1}{C_1} \dot{e}_5 \end{aligned} \quad (3.8)$$

(3.8)式を(3.6)式に代入すると(3.9)式、(3.10)式になる。また、(3.9)式、(3.10)式を一般的な式の形にすると(3.11)式になる。(3.9)式及び(3.10)式中の#はプログラム内において微分記号を表すために用いる文字である。#の次の数字が微分階数を表す。

$$f2\#1=-L1e5+L1E1t+R1L1F1t-R1L1f2 \quad (3.9)$$

$$e5\#1=C1f2-C1F1t \quad (3.10)$$

$$\begin{aligned} \dot{f}_2 &= -L_1 e_5 + L_1 E_1(t) + R_1 L_1 F_1(t) - R_1 L_1 f_2 \\ \dot{e}_5 &= C_1 f_2 - C_1 F_1(t) \end{aligned} \quad (3.11)$$

最後に、(3.9)式に(3.10)式を代入したものが図 3.1 のボンドグラフにおける一般的な支配方程式であり、それを以下に示す。(3.12)式を一般的な式の形にすると(3.13)式になる。

$$f2\#1=-L1C1\$f2d+L1C1\$F1td+L1E1t+R1L1F1t-R1L1f2 \quad (3.12)$$

$$\dot{f}_2 = -L_1 C_1 \int f_2 dt + L_1 C_1 \int F_1(t) dt + L_1 E_1(t) + R_1 L_1 F_1(t) - R_1 L_1 f_2 \quad (3.13)$$

3.2.2 機械系の支配方程式の作成

図 3.1 のボンドグラフを機械系のモデルで表すと図 3.3 の質量、バネ、ダンパー系のモデルになる。機械系においてフローは速度である。(3.12)式の速度を変位に変換したものを(3.15)式に示す。(3.15)式を一般的な式の形にすると(3.16)式になる。速度から変位に変換する際の関係を(3.14)式に示す。

$$\begin{aligned} f_2 &= \dot{x}_2 \\ \int F_1(t) dt &= X_1(t) \end{aligned} \quad (3.14)$$

$$x2\#2=-L1C1x2+L1C1X1t+L1E1t+R1L1F1t-R1L1x2\#1 \quad (3.15)$$

$$\ddot{x}_2 = -L_1 C_1 x_2 + L_1 C_1 X_1(t) + L_1 E_1(t) + R_1 L_1 F_1(t) - R_1 L_1 \dot{x}_2 \quad (3.16)$$

機械系では一般に、質量を m 、ばね係数を K 、ダンパー粘性係数を C 、力を F 、速度を V で表す。従って、(3.17)に示す手順で(3.15)式を変形すると(3.18)式に示す支配方程式が導かれる。(3.18)式を一般的な式の形にすると(3.19)式になる。

$$\frac{1}{L_1} \rightarrow m_1, C_1 \rightarrow K_1, R_1 \rightarrow C_1, E_1 \rightarrow F_1, F_1 \rightarrow V_1 \quad (3.17)$$

$$m1x2\#2=-K1x2+K1X1t+F1t+C1V1t-C1x2\#1 \quad (3.18)$$

$$m_1 \ddot{x}_2 = -K_1 x_2 + K_1 X_1(t) + F_1(t) + C_1 V_1(t) - C_1 \dot{x}_2 \quad (3.19)$$

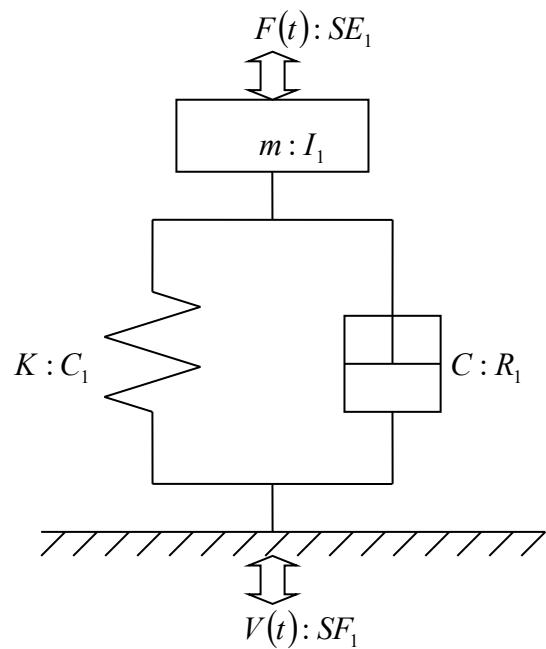


Fig 3.3 Physical Model(Spring-Mass-Damper)

3.2.3 電気系の支配方程式の作成

ここでは図 3.4 の LCR 回路のモデルを用いて説明する. 図 3.4 をボンドグラフで表したもののが図 3.5 である.

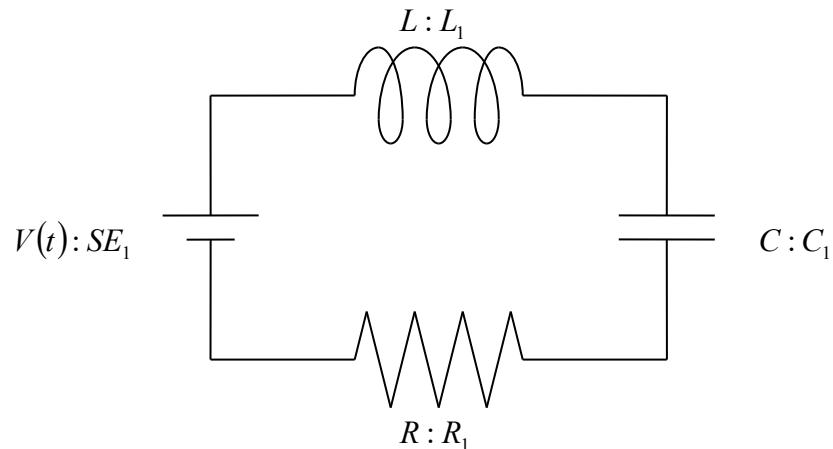


Fig 3.4 Physical model(LCR circuit)

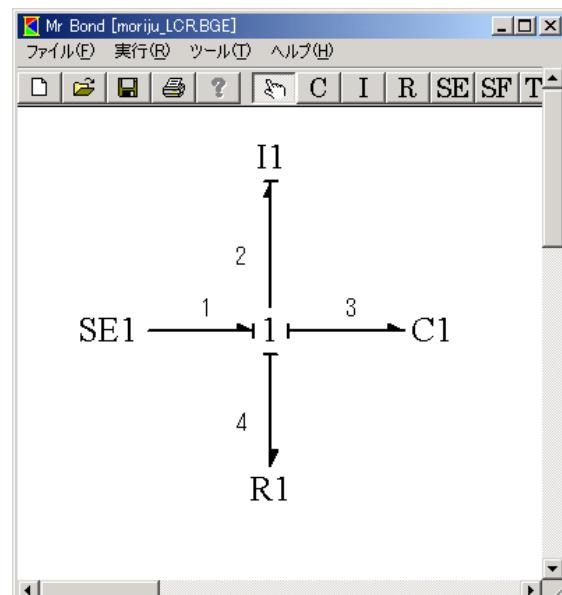


Fig 3.5 Bondgraph model(LCR circuit)

図 3.5 について(3.1)~(3.12)と同様の作業を行うと以下に示す式になる. (3.20)式を一般的な式の形になると(3.21)式になる.

$$f2\#1=L1E1t-R1L1f2-L1C1\$f2d \quad (3.20)$$

$$\dot{f}_2 = L_1 E_1(t) - R_1 L_1 f_2 - L_1 C_1 \int f_2 dt \quad (3.21)$$

電気系においてフローは電流である. (3.20)式のフローを電流を表す i に変換した式を以下に示す. (3.22)

式を一般的な形の式にすると(3.23)式になる.

$$i_2 \# 1 = L_1 E_1 t - R_1 L_1 i_2 - L_1 C_1 \$ i_2 \quad (3.22)$$

$$\dot{i}_2 = L_1 E_1(t) - R_1 L_1 i_2 - L_1 C_1 \int i_2 dt \quad (3.23)$$

電気系では一般に、電圧を V で表現する。従って、(3.24)に示す手順で(3.22)式、(3.23)式で示される式を変形し、1階微分すると(3.25)式に示す支配方程式式が導かれる。(3.25)式を一般的な式の形にすると(3.26)式になる。

$$L_1 \rightarrow \frac{1}{L_1}, C_1 \rightarrow \frac{1}{C_1}, E_1 \rightarrow V_1 \quad (3.24)$$

$$L_1 i_2 \# 2 = V_1 \# 1 t - R_1 i_2 \# 1 - / C_1 i_2 \quad (3.25)$$

$$L_1 \ddot{i}_2 = V_1(t) - R_1 \dot{i}_2 - \frac{1}{C_1} i_2 \quad (3.26)$$

3.2.4 支配方程式の表示

作成した支配方程式を表示する手順を以下に示す。図 3.6 は支配方程式を表示する手順のフローチャートである。分数の場合は分子、分母のそれぞれについて表示される領域の左上の座標を与え、再起を用いて表示する。

1. 作成した支配方程式の文字列を 1 文字ずつ処理する
2. 文字が画像として登録した文字であるかどうか判定する
3. 画像として登録した文字であれば以下の作業を行う
 - 画像をロードする
 - 次の画像を表示するためのオフセットを設定する
 - その画像の添え字の位置を設定する
 - ロードした画像を表示する
4. 画像として登録した文字でなければそれを添え字と判断し、設定した位置に従い文字を表示する
5. 2~4 をすべての文字について行う

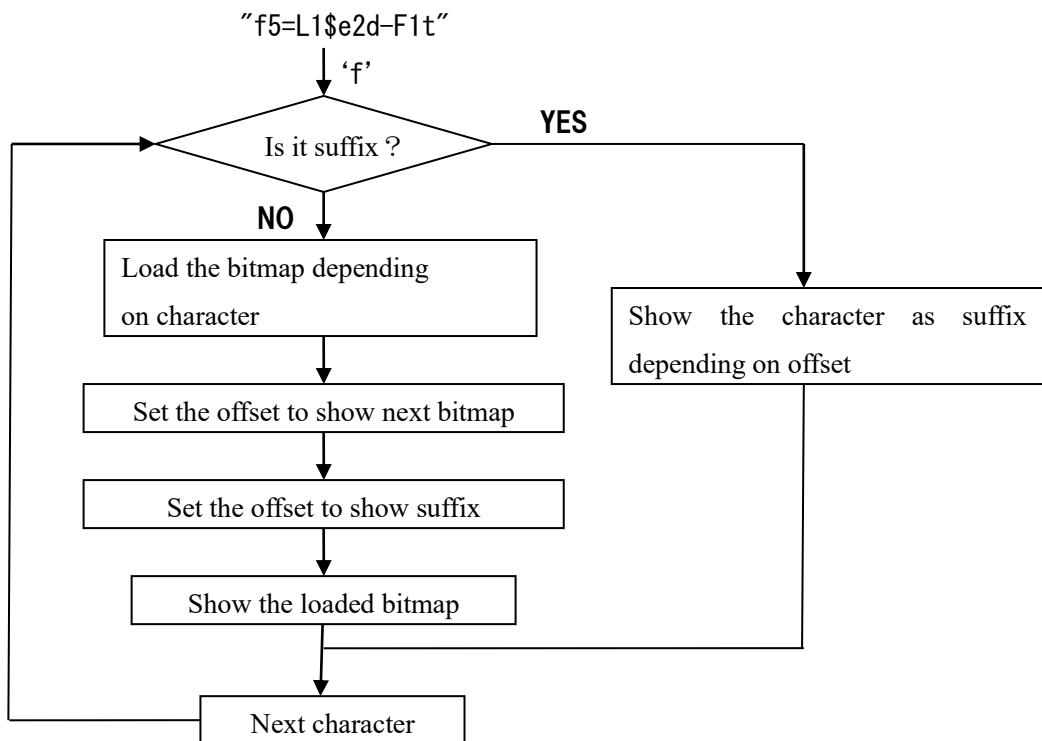


Fig 3.6 Flow chart to show governing equation

図 3.6 のように以下の文字列を表示する場合を例にとり手順を説明する。

$$f5=L1$e2d-F1t \quad (3.27)$$

1. まず、'f' の文字について処理を行う

2. 'f' は画像であると登録しているので以下の処理を行う
 - (ア) f の画像をロードする
 - (イ) 次の画像のオフセットを設定する
 - (ウ) f の画像における添え字の位置を設定する
 - (エ) f の画像を表示する
3. 次の文字の'5'について処理を行う
4. '5' は画像であると登録していないので、添え字であると判断する
5. '5'を2で設定した位置に従って文字として表示する
6. '=' 以降の文字についても同様に作業を行う
7. '\$'の文字の場合は以下の処理を行う
 - (ア) 積分記号である \int の画像をロードする
 - (イ) 次の画像のオフセットを設定する
 - (ウ) 積分記号である \int の画像を表示する
8. 'd' の文字の場合は以下の処理を行う
 - (ア) dt の画像をロードする
 - (イ) 次の画像のオフセットを設定する
 - (ウ) dt の画像を表示する
9. 't' の文字の場合は以下の処理を行う
 - (ア) (t) の画像をロードする
 - (イ) 次の画像のオフセットを設定する
 - (ウ) (t) の画像を表示する

以上に示した手順でウィンドウに表示させると以下になる。

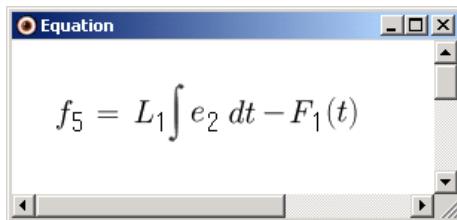


Fig 3.7 Display equation on window

3.3 実行結果

3.3.1 機械系の実行結果

機械系の支配方程式を開発したコードを用いて表示させ、その妥当性を検証する。図 3.8 は機械系のモデルである。

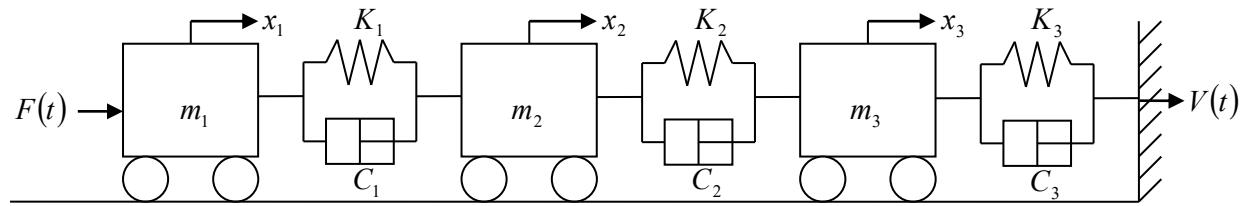


Fig 3.8 Physical model(Spring-Mass-Damper)

このシステムの支配方程式を以下に示す.

$$\begin{aligned}
 m_1 \ddot{x}_1 + C_1(\dot{x}_1 - \dot{x}_2) + K_1(x_1 - x_2) &= F(t) \\
 m_2 \ddot{x}_2 - C_1(\dot{x}_1 - \dot{x}_2) + C_2(\dot{x}_2 - \dot{x}_3) - K_1(x_1 - x_2) + K_2(x_2 - x_3) &= 0 \\
 m_3 \ddot{x}_3 - C_2(\dot{x}_2 - \dot{x}_3) + C_3(\dot{x}_3 - V(t)) - K_2(x_2 - x_3) + K_3(x_3 - \int V(t) dt) &= 0
 \end{aligned} \tag{3.28}$$

また、図 3.8 のモデルからボンドグラフを作成すると以下になる.

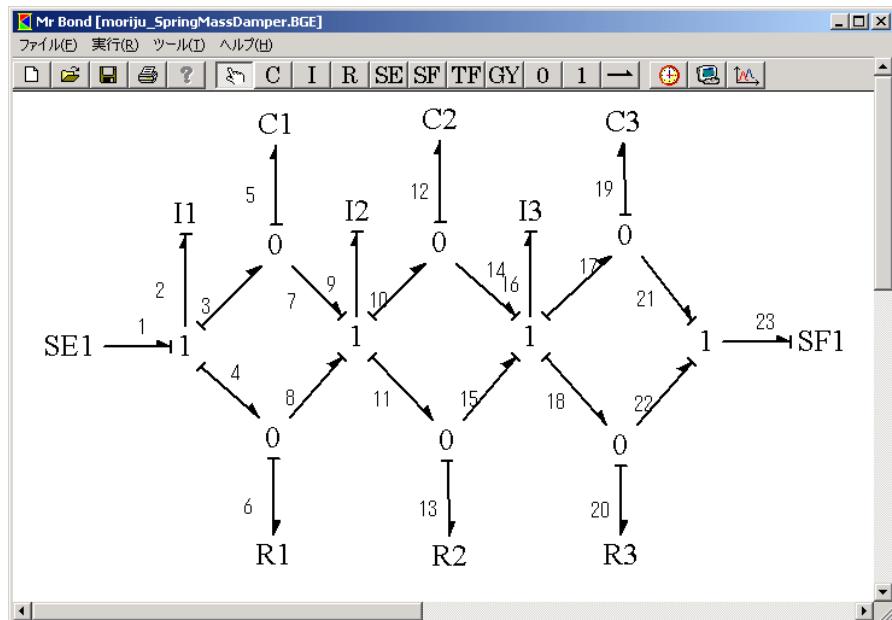


Fig 3.9 Bondgraph model(Spring-Mass-Damper)

さらに、図 3.9 をもとに開発したコードを用いて支配方程式を表示をさせると以下になる.

$$m_1 \ddot{x}_2 = F_1(t) + K_1 x_9 - K_1 x_2 - C_1 \dot{x}_2 + C_1 \dot{x}_9$$

$$m_2 \ddot{x}_9 = -K_1 x_9 + K_1 x_2 + K_2 x_{16} - K_2 x_9 - C_2 \dot{x}_9 + C_2 \dot{x}_{16} + C_1 \dot{x}_2 - C_1 \dot{x}_9$$

$$m_3 \ddot{x}_{16} = -K_2 x_{16} + K_2 x_9 + K_3 X_1(t) - K_3 x_{16} - C_3 \dot{x}_{16} + C_3 V_1(t) + C_2 \dot{x}_9 - C_2 \dot{x}_{16}$$

Fig 3.10 Governing equation(Spring-Mass-Damper)

(3.28)式における変数と、図 3.10 における変数との対応関係を以下に示す。左辺が(3.28)式、右辺が図 3.10 である。

$$\begin{aligned} x_1 &= x_2 \\ x_2 &= x_9 \\ x_3 &= x_{16} \\ F(t) &= F_1(t) \\ \int V(t) dt &= X_1(t) \end{aligned} \tag{3.29}$$

(3.29)式を(3.28)式に代入し、変形すると以下になる。(3.30)式は図 3.10 に示す支配方程式と同じである。

$$\begin{aligned} m_1 \ddot{x}_2 &= F_1(t) + K_1 x_9 - K_1 x_2 - C_1 \dot{x}_2 + C_1 \dot{x}_9 \\ m_2 \ddot{x}_9 &= -K_1 x_9 + K_1 x_2 + K_2 x_{16} - K_2 x_9 - C_2 \dot{x}_9 + C_2 \dot{x}_{16} + C_1 \dot{x}_2 - C_1 \dot{x}_9 \\ m_3 \ddot{x}_{16} &= -K_2 x_{16} + K_2 x_9 + K_3 X_1(t) - K_3 x_{16} - C_3 \dot{x}_{16} + C_3 V_1(t) + C_2 \dot{x}_9 - C_2 \dot{x}_{16} \end{aligned} \tag{3.30}$$

従って、図 3.10 に示す支配方程式は(3.28)式と同じであり、開発したコードは機械系の支配方程式を正しく表示していることがわかる。

3.3.1 電気系の実行結果

次に、電気系の支配方程式を開発したコードを用いて表示させ、その妥当性を検証する。図 3.11 は電気系のモデルある。

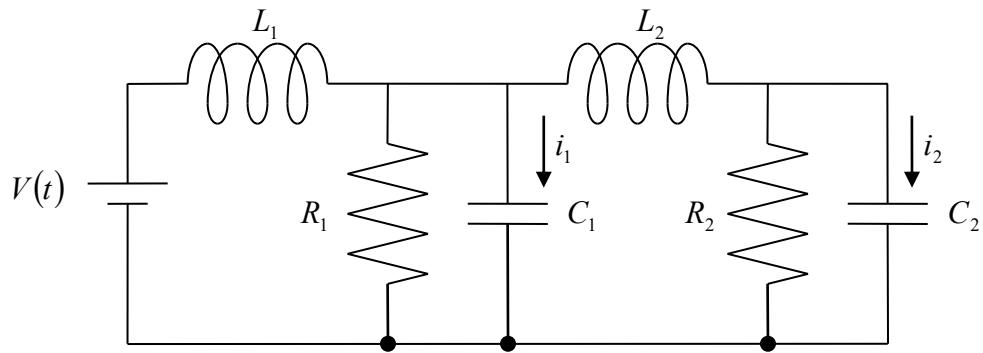


Fig 3.11 Physical model(LCR circuit)

このシステムの支配方程式を以下に示す.

$$\begin{aligned}
 L_1 \ddot{i}_1 &= \dot{V}(t) - \frac{L_1}{R_1 C_1} i_1 - \frac{1}{C_1} i_1 - \frac{L_1}{L_2 C_1} i_1 + \frac{L_1}{L_2 C_2} i_2 \\
 L_2 \ddot{i}_2 &= -\frac{L_2}{R_2 C_2} i_2 + \frac{1}{C_1} i_1 - \frac{1}{C_2} i_2
 \end{aligned} \tag{3.31}$$

また、図 3.11 からボンドグラフを作成すると以下になる.

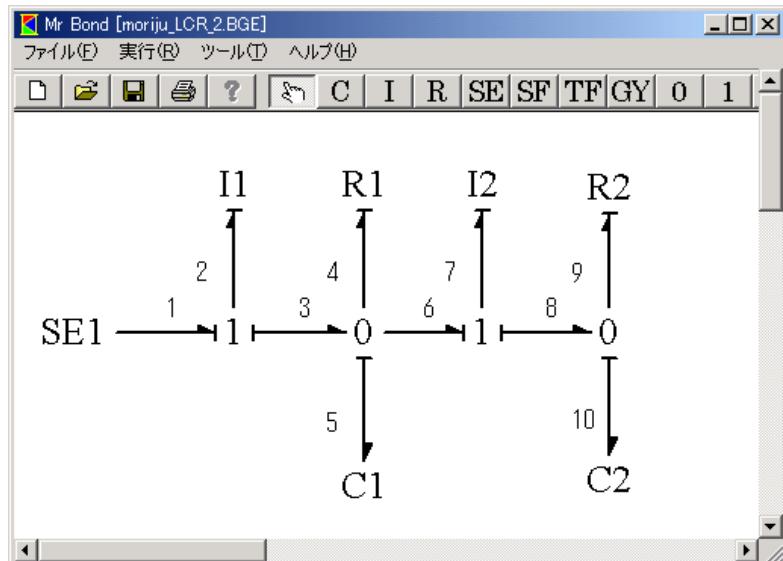


Fig 3.12 Bondgraph model(LCR circuit)

さらに、図 3.12 をもとに開発したコードを用いて支配方程式を表示させると以下になる.

Fig 3.13 Governing equation(LCR circuit model)

(3.31)式における変数と、図 3.13 における変数の対応関係を以下に示す。左辺が(3.31)式、右辺が図 3.13 である。

$$\begin{aligned} i_1 &= i_5 \\ i_2 &= i_{10} \\ V(t) &= V_1(t) \end{aligned} \tag{3.32}$$

(3.32)式を(3.31)式に代入すると以下になる。(3.33)式は図 3.13 に示す支配方程式と同じである。

$$\begin{aligned} \ddot{i}_5 &= \frac{1}{L_1} \dot{V}_1(t) - \frac{1}{L_1 C_1} i_5 - \frac{1}{R_1 C_1} \dot{i}_5 - \frac{1}{L_2 C_1} i_5 + \frac{1}{L_2 C_2} i_{10} \\ \ddot{i}_{10} &= \frac{1}{L_2 C_1} i_5 - \frac{1}{L_2 C_2} i_{10} - \frac{1}{R_2 C_2} \dot{i}_{10} \end{aligned} \tag{3.33}$$

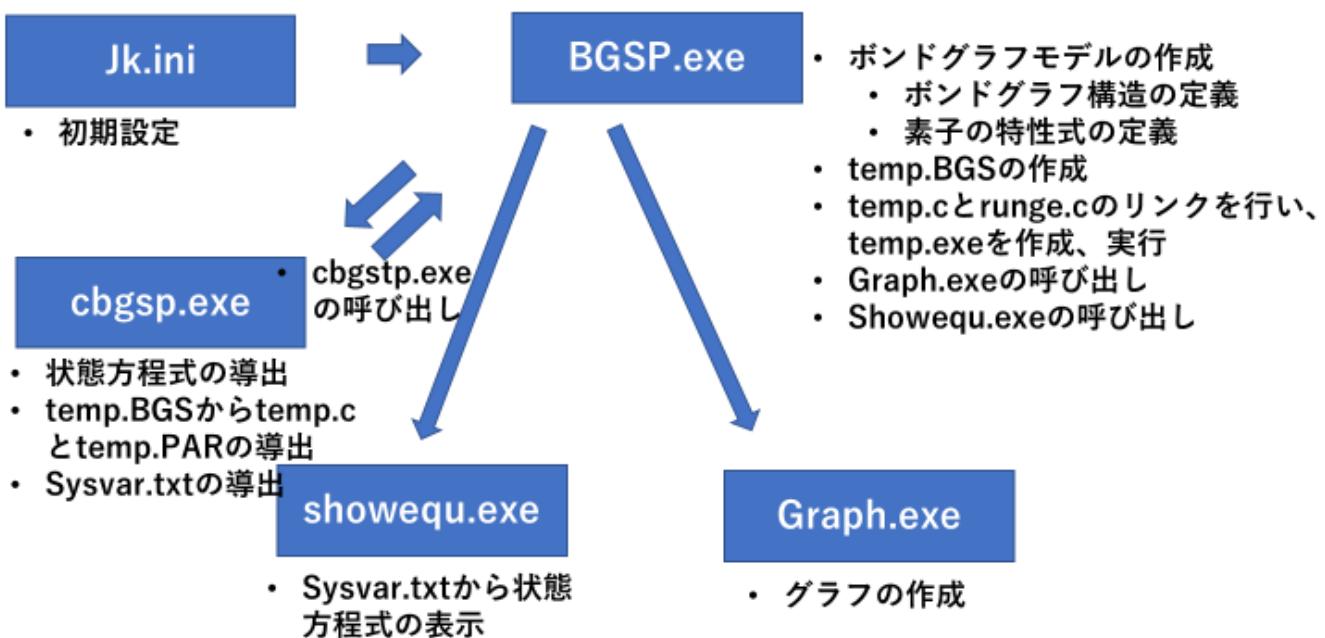
従って、図 3.13 に示す支配方程式と(3.31)式は同じであり、開発したコードは電気系の支配方程式を正しく表示していることがわかる。

付録. 4 BGSPVer.2.0 以降のファイル構造について

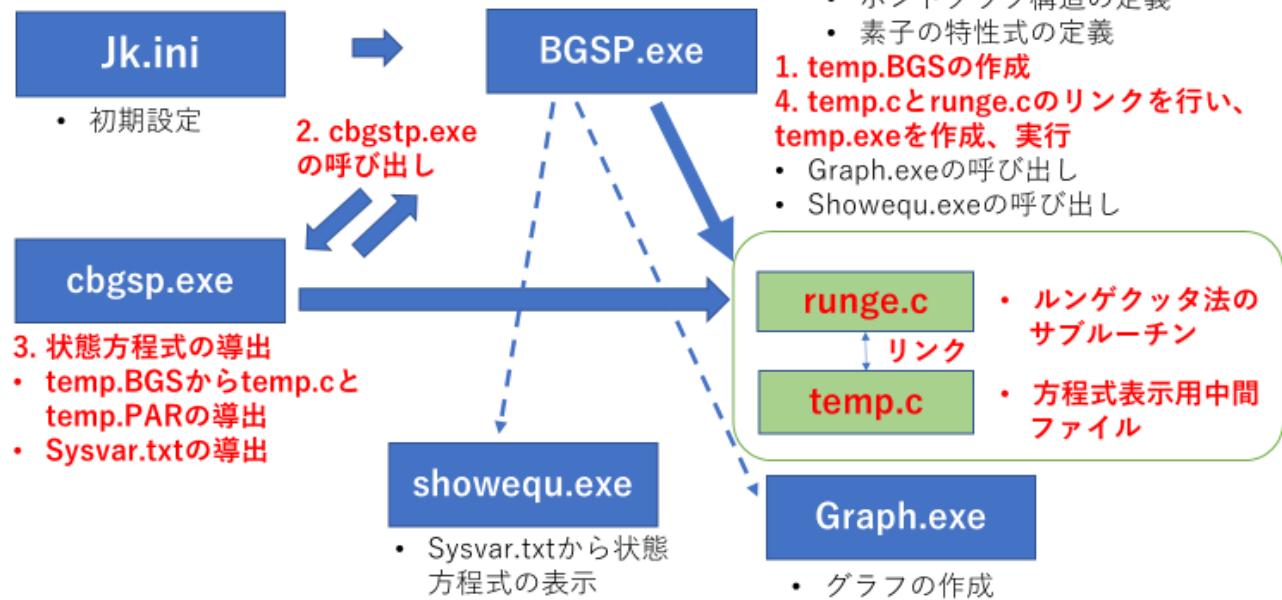
BGSPVer.2の基本的なファイル構成と意味

BGSP.exe	<ul style="list-style-type: none">・ ポンドグラフモデルの作成<ul style="list-style-type: none">・ ポンドグラフ構造の定義・ 素子の特性式の定義・ temp.BGSの作成・ cbgsp.exeの呼び出し・ temp.cとrunge.cのリンクを行い、temp.exeを作成、実行	showequ.exe	<ul style="list-style-type: none">・ Sysvar.txtから状態方程式の表示
		Jk.ini	<ul style="list-style-type: none">・ 初期設定
cbgsp.exe	<ul style="list-style-type: none">・ 状態方程式の導出・ temp.BGSからtemp.cとtemp.PARの導出・ Sysvar.txtの導出	Graph.exe	<ul style="list-style-type: none">・ グラフの作成
Runge.c	<ul style="list-style-type: none">・ ルンゲクッタ法のサブルーチン	temp.c	<ul style="list-style-type: none">・ 実行用中間ファイル
		sysvar.txt	<ul style="list-style-type: none">・ 方程式表示用中間ファイル

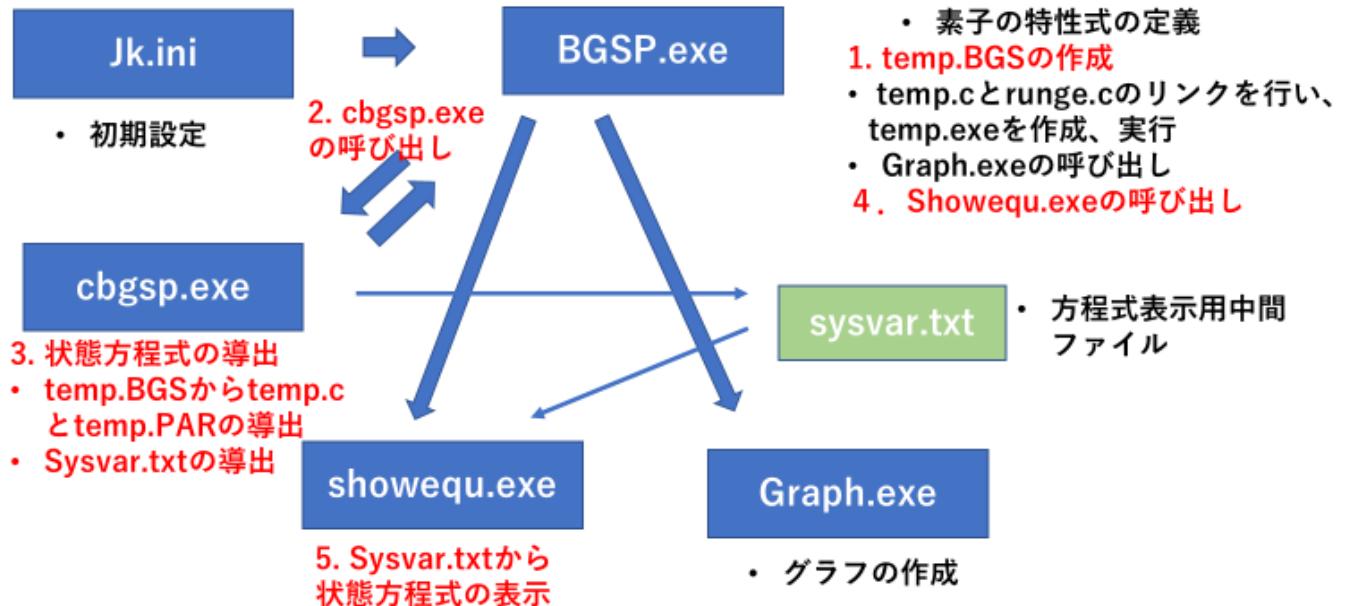
BGSPVer.2のファイルの関係



計算実行時の流れ



状態方程式導出時の流れ



- ボンドグラフモデルの作成
 - ボンドグラフ構造の定義
 - 素子の特性式の定義
- 1. temp.BGSの作成
- 4. temp.cとrunge.cのリンクを行い、temp.exeを作成、実行
 - Graph.exeの呼び出し
 - Showequ.exeの呼び出し

- ルンゲクッタ法のサブルーチン
- 方程式表示用中間ファイル

付録. 5 BGSPVer.2.0 以降のデバッグについて

BGSP の仕様に際し、実行ファイルが生成されないなどの不具合が発生した場合、以下にデバッグの手順をフローチャートとして示す。

BGSP Ver.2のデバッグ方法



BGSP(Bond Graph Simulation Program)は, 1985 年に, 日本機械学会研究協力部会, RC68 油空圧システム省エネルギー調査研究分科会の成果として報告されたものである。BGSPは, 当時, 通商産業省工業技術院機械技術研究所(現, 産業技術総合研究所)で, 幸田武久博士(機械技研→京都大学, 故人), 中田毅博士(機械技研→東京電機大学, 名誉教授)を中心に製作され, 一般に公開(1988年)されているシミュレーションプログラムである。その後に, (社)日本油空圧学会(現, 一般社団法人, 日本フルードパワーシステム学会)の OHC-Sim 委員会(桜井康雄博士(足利大学))で, 油圧記号のアイコン化とマッチさせて油空圧分野でより使いやすくするためにプログラムの修正なども行われた。

BGSP には, ボンドグラフで表されるシステム変数間の因果関係に着目した, 簡単な数式処理を用いることにより,

- ① システム状態方程式が, 関数表現形で自動的に導出されてシミュレーションが行われ,
 - ② 非線形特性を持つ要素を含むシステムが簡単に取り扱え,
 - ③ フィードバック特性を持つような素子の特性が使用者により自由に定義でき,
 - ④ システム動特性のほかに使用者が定義するエネルギー効率などの特性値が計算できる,
- などの特徴がある。プログラムは FORTRAN 77 言語で記述されており, パソコン上でも使用でき, 他のシステムへの移植も簡単である。

上記の紹介文は, J.U. Thoma & 須田信英 著, ボンドグラフによるシミュレーション, コロナ社, 1996年, をほとんど原文のまま, ただし, 誤った記述を修正するなどを行って後に掲載したものである。BGSP の特徴がよく明示されている。

その後, BGSP は C 言語, C++ 言語でも書き下されて, BGSP Ver2 は双方の言語での使用が可能である。また, 当初の BGSP(便宜上, これを Ver.1 と呼ぶ)では, システム・ボンドグラフ構造の入力形式がプログラム・ベースであったために, 使用方法が煩雑であり一般化しなかつた。そこで, 富岡弘毅博士(九州工業大学, 博士修了, 故人), 田中和博博士(九州工業大学, 名誉教授)により GUI を利用して画面上でボンドグラフ構造の入力が可能となった。

その後, 肥後寛氏(九州工業大学)の協力を得て, プログラムのデバッグが行われ修正された箇所がある。またいくつかの便利な機能が付加, 拡充された。そこで, BGSP Ver.2 の特徴を纏めると, 上記の①②③④に下記の項目を付け加えることとなる。

- ⑤ 入力素子に実験値入力が可能であり,
- ⑥ 変数のチェック機能が充実され,
- ⑦ 何にでも利用できる便利なグローバル変数 $XX(i)$ が用意されたことで, 再帰型ループの解消が容易に可能となっており,
- ⑧ システム・ボンドグラフより状態方程式の導出が可能である。

BGSP Ver.2 は各種の機能が整うまでに時間が掛かりましたが, GUI は15年前にほぼ完成しており, それを利用したシステム動特性解析はこれまでに200例を超えます。これらの結果の中で不合理な結果を得ることはなかったと考えています。したがって, BGSP Ver.2 を信頼して利用していただければ幸いと考えています。しかし残念ながら, 気が付かなかった間違いもあるものと思われますので, もしお気づきの点があれば是非ご連絡を願いいたします。

(文責, 田中和博)