

ModelNgspicer User Guide

Version 1.0.0

Developed by 株式会社

1 はじめに

ModelNgspicer は、ngspice をベースにした GUI アプリケーションであり、回路設計の最適化やデバイスモデリングにおける作業効率の向上を目的としています。GUI は Python の Qt ライブラリ (PySide6) を用いて構築されており、マウスホイールや数値入力によってモデルパラメータを調整し、特性の変化をリアルタイムでグラフ表示することが可能です。

Ngspice は、オープンソースの SPICE 系回路シミュレータであり、DC 解析、AC 解析、過渡解析、ノイズ解析、S パラメータ解析など、多彩な解析機能を備えています。また、Verilog-A や BSIM などの業界標準モデルにも対応しており、高精度なデバイスモデリングが可能です。コマンドラインベースで動作するため、スクリプトによる複雑な計算処理や自動解析にも適しています。

Ngspice

<https://ngspice.sourceforge.io>

2 インストール

2.1 動作環境

ModelNgspicer は、以下の環境で動作確認されています。

- OS : Windows 10 以降 (64 ビット)
- Python : バージョン 3.13 以上
- 必要なライブラリ : PySide6、PyQtGraph、numpy
- Ngspice : バージョン 43 以上を推奨

2.2 インストール手順 (Windows)

Windows 環境でのインストール手順を以下に示します。ここでは、Windows 用パッケージマネージャ「Chocolatey」を用いたインストールを推奨しますが、公式サイトからダウンロードしてインストールしても OK です。その場合は、インストール後、環境変数にパスが通っていることを確認してください。

1. Chocolatey のインストール

管理者権限で PowerShell を開き、以下のコマンドを実行します。詳細は Chocolatey の公式ページ (<https://chocolatey.org/install>) を参照してください。

Powershell

```
Set-ExecutionPolicy Bypass -Scope Process -Force;
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object
System.Net.WebClient).DownloadString('https://community.chocolatey.org/
install.ps1'))
```

2. Python のインストール

Chocolatey を用いて Python をインストールします。管理者権限の PowerShell またはコマンドプロンプトで以下のコマンドを実行します。インストール後、`python` および `pip` コマンドが使用可能であることを確認してください。

Powershell
<code>choco install python</code>

4. ngspice のインストール

Chocolatey を用いて ngspice をインストールします。管理者権限の PowerShell またはコマンドプロンプトで以下のコマンドを実行します。インストール後、`ngspice` および `ngspice_con` コマンドが使用可能であることを確認してください。

Powershell
<code>choco install ngspice</code>

3. 必要ライブラリのインストール

以下のコマンドで、必要な Python ライブラリをインストールします。

Powershell
<code>pip install pyside6 pyqtgraph numpy</code>

以上で、セットアップは完了です。プロジェクトフォルダ内の `run.vbs` を実行することで、ModelNgspicer を起動できます。

3 GUI と基本操作

3.1 画面構成の概要

以下の図は、ModelNgspicer の画面構成を示しています。

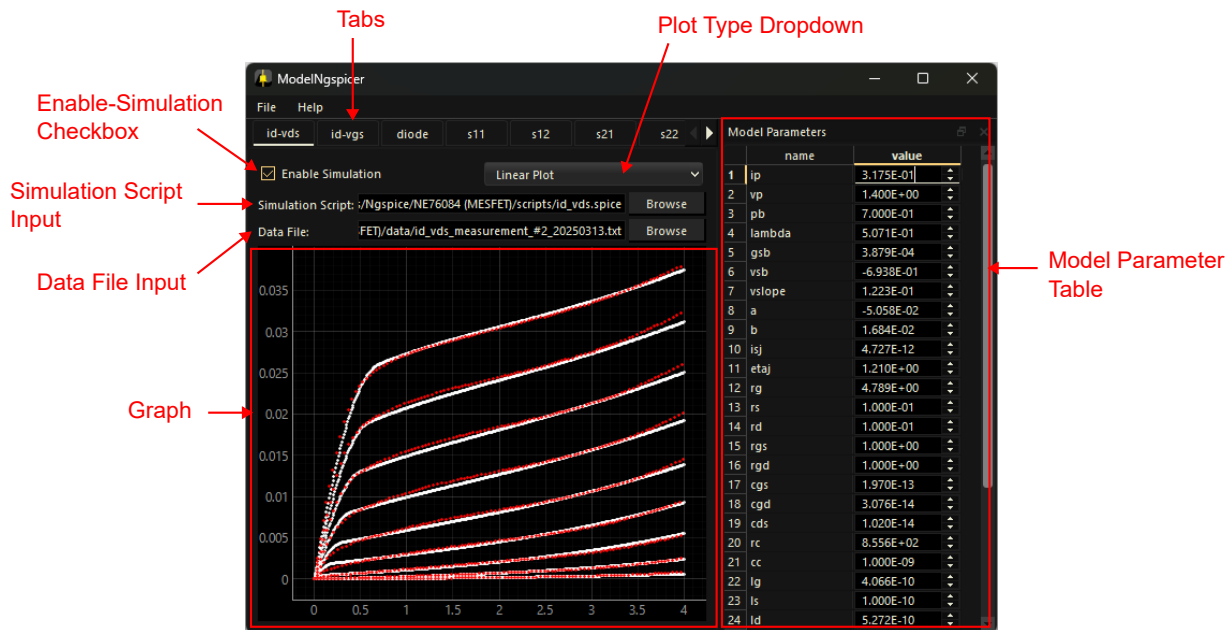


Fig. 3-1 GUI 画面の構成

- **タブパネル (Tabs)**

画面上部には 10 個のタブが並んでおり、それぞれ異なるシミュレーション設定を保持できます。タブ名はダブルクリックで変更可能です。右クリックから「Pop out」を選択すると、タブの内容を別ウィンドウとして表示できます。ポップアウトしたウィンドウは、閉じると元の位置に戻り、複数のウィンドウを同時に開いて平行してパラメータ調整を行うことも可能です。

- **モデルパラメータテーブル (Model Parameter Table)**

モデルパラメータの一覧が表示されます。メニューバーの「File > Load Parameters...」からパラメータ定義ファイルを読み込むことで、テーブルに反映されます。各パラメータの値はスピンドルボックスに表示され、カーソルを合わせてマウスホイールを操作することで値を変更できます。Ctrl キーを押しながら操作すると、粗い調整が可能です。

- **シミュレーション有効化チェックボックス (Enable Simulation Checkbox)**

現在表示されているタブのシミュレーションを有効化／無効化できます。有効化されている場合、パラメータを変更するたびにシミュレーションが実行されます。不要なタブのシミュレーションを無効化しておくことで、実行時間を短縮できる可能性があります。

- **シミュレーションスクリプト入力 (Simulation Script Input)**

ngspice 用のスクリプトファイルを選択するフィールドです。「Browse」ボタンを押すとファ

イルダイアログが開き、スクリプト（**.spice**、**.sp**、**.cir**）を選択できます。選択されたスクリプトは Python から **ngspice_con** コマンドで実行され、同名の **.txt** ファイルとして結果が出力されます。この結果は自動的にグラフ領域にプロットされます。

- **データファイル入力（Data File Input）**

シミュレーション結果との比較用に、外部データファイルを読み込むことができます。読み込まれたデータは、シミュレーション結果と同じグラフ上にプロットされます。測定結果やデータシートの読み値、目標値などを重ねて確認したい時に便利です。

- **プロット形式選択（Plot Type Dropdown）**

グラフの表示形式を選択します。Linear Plot、Log-Log、Semi-Log X、Semi-Log Y、Smith Chart のの中から選択できます。

- **グラフ表示領域（Graph）**

シミュレーション結果は白点、データファイルの値は赤点でプロットされます。

3.2 基本操作フロー

ModelNgspicer の基本的な使用手順は以下の通りです。

1. モデルパラメータの読み込み
メニューバーの「File > Load Parameters...」をクリックし、パラメータ定義ファイル（.txt）を選択します。読み込まれたパラメータは、モデルパラメータテーブルに表示されます。
2. シミュレーションスクリプトの指定
シミュレーションスクリプト入力の「Browse」ボタンをクリックし、使用する ngspice スクリプト（.spice、.sp、.cir）を選択します。
3. シミュレーションの有効化
シミュレーション有効化チェックボックスをオンにすると、現在のタブに対してシミュレーションが有効になります。パラメータを変更するたびにシミュレーションが実行されます。
4. パラメータの調整
対応するスピンボックスを操作することでモデルパラメータを変更します。
 - マウスホイール操作で微調整
 - Ctrl キーを押しながらマウスホイール操作で粗調整
5. 比較データの読み込み（任意）
データファイル入力の「Browse」ボタンから外部データファイル（測定結果や目標値など）を指定できます。読み込まれたデータは、シミュレーション結果と同じグラフ上に重ねて表示されます。
6. プロット形式の選択
プロット形式選択のドロップダウンリストから、グラフの表示形式を選択します。Linear、Log-Log、Semi-Log X、Semi-Log Y、Smith Chart のの中から選択できます。
7. 結果の確認
グラフ領域において、シミュレーション結果は白点、外部データは赤点でプロットされます。パラメータを調整すると、グラフがリアルタイムで更新されます。
8. モデルパラメータの保存
調整後のモデルパラメータを保存するには、メニューバーの「File > Save Parameters...」を選択します。

4 ファイル形式

4.1 パラメータ定義ファイル

ModelNgspicer では、モデルパラメータをプレーンテキストファイル（.txt）で定義します。ファイル内の各行は、以下のように先頭に+記号を付けて記述します。

Plain Text

```
+ is = 1e-14
+ n = 1.5
+ rs = 0.5
+ ...
```

この+記号は SPICE 構文における行継続記号であり、前の行からの継続を意味します。この形式を採用することで、.param や .model ステートメントにそのまま展開可能となり、ngspice スクリプトへの組み込みが容易になります。

例 1 .param 文に展開する場合：

Ngspice Script

```
.param
.include model.txt
```

上のスクリプトのように .param の直下に .include 文を記述すると、ngspice は内部的に以下のように展開します：

Ngspice Script

```
.param
+ is = 1e-14
+ n = 1.5
+ rs = 0.5
+ ...
```

すなわち、パラメータファイルの内容が .param の引数として扱われ、ngspice スクリプトに簡潔に埋め込むことができます。

例 2 .model 文に展開する場合：

Ngspice Script

```
.model DIODE1 D (
.include model.txt
+ )
```

上の例では、パラメータファイルの内容が **DIODE1** のモデルパラメータとして適用されます。このように、カスタムデバイスモデルを作成することができます。

4.2 シミュレーション結果とデータファイル

ModelNgspicer では、シミュレーション結果ファイルと外部データファイルの両方に、共通のプレーンテキスト形式 (**.txt**) を使用します。

ファイル構造：

- ヘッダー行なし
- 1 列目：X 軸データ
- 2 列目以降：Y 軸データ
- 区切り文字：スペースまたはタブ

例：

```
Plain Text
1e6 -3.2 -1.1
2e6 -3.5 -1.3
3e6 -3.8 -1.6
...
```

シミュレーション結果の出力方法：

ngspice スクリプト内で **wrdata** コマンドを使用することで、シミュレーション結果を出力できます。出力ファイル名は、スクリプトファイルと同じ名前+**.txt** 拡張子にする必要があります。この命名規則により、ModelNgspicer が自動的に結果ファイルを認識し、グラフにプロットします。

出力例 (ngspice スクリプト)：

```
Ngspice Script
set wr_singlescale
wrdata iv.txt i(v1) i(v2)
```

この例では、**iv.spice** というスクリプトを実行すると、**iv.txt** という結果ファイルが生成され、**i(v1)** と **i(v2)** の電流値が記録されます。

注意点：

- ファイル名の一致は必須です。スクリプト名と結果ファイル名が一致していない場合、自動プロットが行われません。
- データの整合性が重要です。不正な形式や不整合のあるデータが含まれていると、グラフが正しく描画されない可能性があります。。

5 活用例

5.1 LC バンドパスフィルタの設計

この節では、ModelNgspicer を用いた LC バンドパスフィルタの設計例を紹介します。Fig. 5-1 に回路図を示します。ノード名を `n01`~`n05` としてラベル付けしており、モデルパラメータ `Cval1`、`Lval2`、`Cval3`、`Lval4` を使用します。

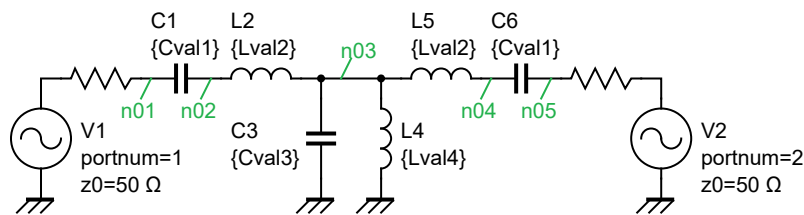


Fig. 5-1 LC バンドパスフィルタの回路図

今回は、バンドパスフィルタの中心周波数が 100 MHz となるように設計を進めてみましょう。初期値としてモデルパラメータを以下のように定義します：

ファイル名：`model_initial.txt`

```
+ Cval1=10p
+ Lval2=300n
+ Cval3=253p
+ Lval4=10n
```

次に、ngspice スクリプトを作成します。

ファイル名：`bpf1.spice`

```
1 bpf1.spice
2
3 .param
4 .include model.txt
5
6 V1  n01 0      dc 0 portnum 1 z0 50
7 V2  n05 0      dc 0 portnum 2 z0 50
8 C1  n01 n02    {Cval1}
9 L2  n02 n03    {Lval2}
10 C3  n03 0      {Cval3}
11 L4  n03 0      {Lval4}
12 L5  n03 n04    {Lval2}
13 C6  n04 n05    {Cval1}
14
15 .control
16 sp lin 200 50Meg 150Meg
17
18 set wr_singlescale
```

```

19 wrdata bpf1.txt db(s_2_1) db(s_1_1)
20
21 .endc
22 .end

```

スクリプトの解説：

1 行目：ngspice では 1 行目はコメントとして扱われます（今回はファイル名を記載）。

3 行目：.param 文を開始します。

4 行目：.include model.txt により、.param 文の続きとしてモデルパラメータを展開します。

6～13 行目：回路構成をネットリストで記述しています。電圧源 V1、V2 は portnum を指定することで RF ポートとして扱われ、z0 によって基準インピーダンス（ここでは 50 Ω）を設定しています。

15 行目：.control は制御セクションの開始を示します。制御セクションでは、ngspice におけるバッチ処理を行うことができます。

16 行目：sp lin 200 50Meg 150Meg は S パラメータ解析コマンドです。50 MHz～150 MHz の範囲を 200 ポイントで線形スイープします。

18 行目：set wr_singlescale を指定することで、wrdata の出力形式が、1 列目に X 軸データ、2 列目以降に Y 軸データとなります。

19 行目：wrdata bpf1.txt db(s_2_1) db(s_1_1) は、S21 および S11 のシミュレーション結果を bpf1.txt に出力します。このファイル名は、スクリプトファイルと同じ名前.txt にする必要があります。

21 行目：.endc は制御セクションの終了を示します。

22 行目：.end はスクリプト全体の終了を示します。

以上で準備が整ったので、ModelNgspicer を使用して実際に動かしてみましょう。

メニューバーから「File > Load Parameters...」を選択、先ほど作成した **model_initial.txt** を読み込みます。次に、シミュレーションスクリプト入力の「Browse」ボタンから **bpf1.spice** を選択します。

すると、Fig. 5-2 のような画面となり、グラフには S_{21} と S_{11} の周波数特性がプロットされます。この時点では、モデルパラメータは初期値のままであり、フィルタ特性がややいびつな形になっていることが分かります。そこで、**Cval1** の値を調整して、フィルタの中心周波数を 100 MHz に近づけてみましょう。

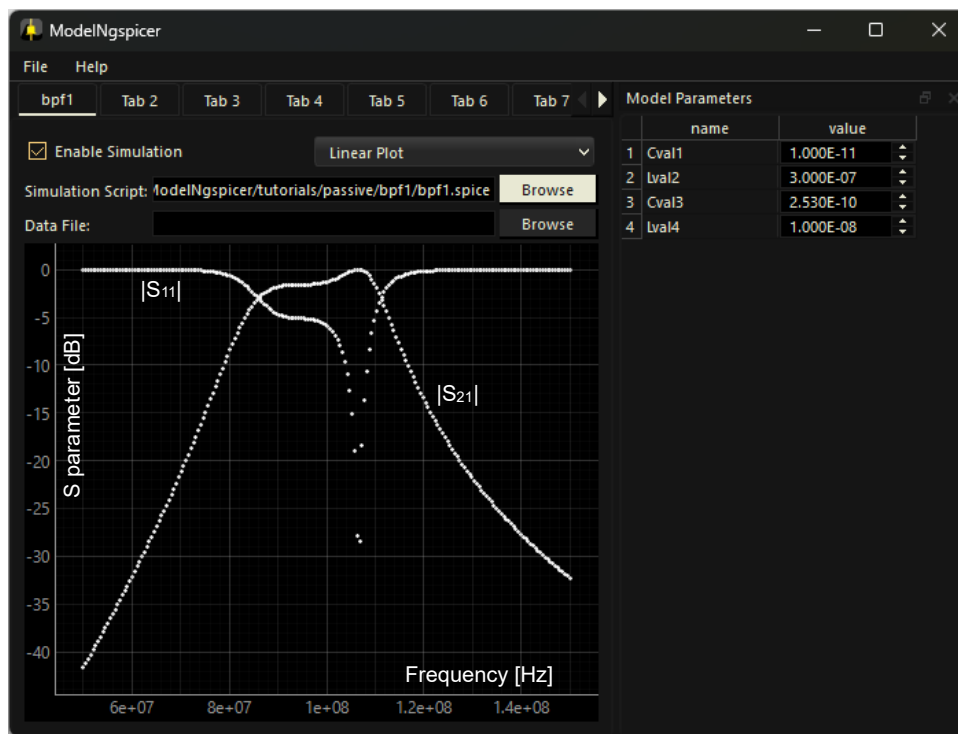


Fig. 5-2 LC バンドパスフィルタの設計（パラメータ調整前）

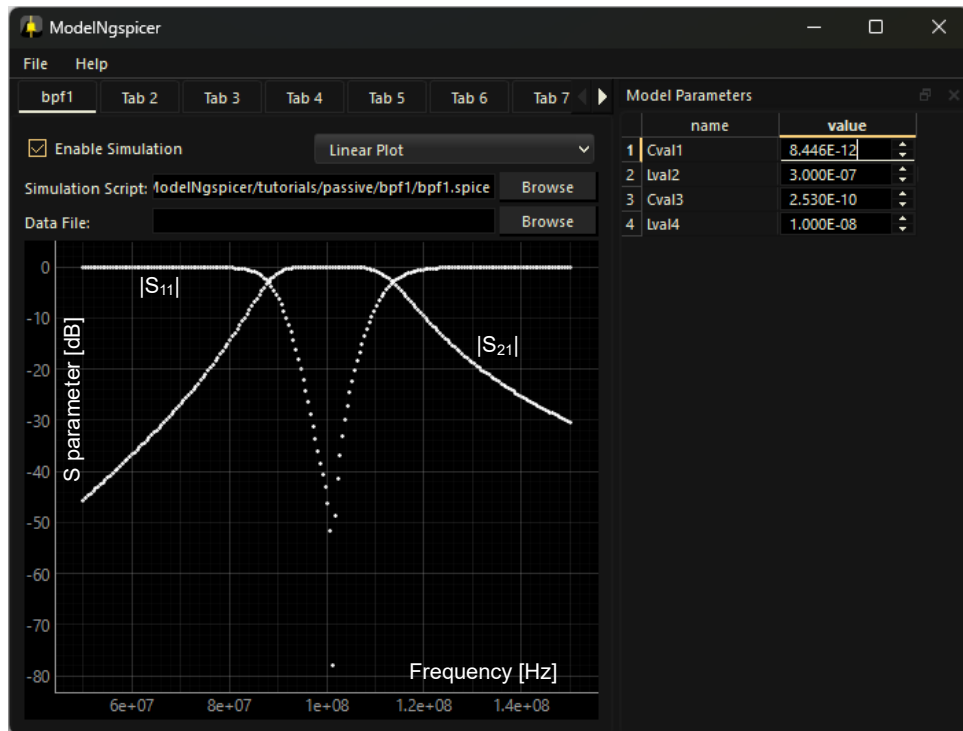


Fig. 5-3 LC バンドパスフィルタの設計（パラメータ調整後）

調整後の結果を Fig. 5-3 に示します。通過帯域内の特性が滑らかになり、中心周波数付近で最大ゲインが得られていることが確認できます。

最後に、調整したモデルパラメータを保存しておきましょう。メニューバーの「File > Save Parameters...」を選択すると、現在のパラメータ値をテキストファイルとして保存できます。

5.2 ダイオードのモデリング

この節では、ModelNgspicer を使用したダイオードのモデリングの例を紹介します。実際に、Onsemi 製 1N4148 のデータシートを基にモデルパラメータを合わせ込んでみましょう。ここでは、以下の特性に着目してモデリングを行います。

- 順方向・逆方向電流-電圧特性 (I_F-V_F , I_R-V_R)
- 接合容量-逆電圧特性 (C_T-V_R)
- 逆回復時間 (t_{rr})

Onsemi, 1N4148

<https://www.onsemi.com/products/discrete-power-modules/small-signal-switching-diodes/1n4148>

5.2.1 モデルパラメータ

以下に、調整対象とする SPICE モデルのパラメータ一覧およびパラメータファイル (`model_final.txt`) を示します。

Junction DC parameters

Name	Parameter	Units	Default	Final value
IS	Saturation current	A	1.0e-14	1.432E-08
JSW	Sidewall saturation current	A	0.0	0.000E+00
N	Emission coefficient	-	1.0	2.346E+00
RS	Ohmic resistance	Ω	0.0	5.506E-01
BV	Reverse breakdown voltage	V	infinity	1.293E+02
IBV	Current at breakdown voltage	A	1.0e-3	5.000E-06
NBV	Breakdown Emission Coefficient	-	N	4.803E+01
IKF	Forward knee current	A	0.0	1.000E-02
IKR	Reverse knee current	A	0.0	1.000E-06
JTUN	Tunneling saturation current	A	0.0	6.750E-11
JTUNSW	Tunneling sidewall saturation current	A	0.0	0.000E+00
NTUN	Tunneling emission coefficient	-	30	5.238E+02
KEG	EG correction factor for tunneling	-	1.0	1.000E+00
ISR	Recombination saturation current	A	1.0e-14	1.639E-10
NR	Recombination current emission coefficient	-	2.0	1.489E+00

Junction capacitance parameters

Name	Parameter	Units	Default	Final value
CJO	Zero-bias junction bottom-wall capacitance	F	0.0	8.685E-13
FC	Coefficient for forward-bias depletion bottom-wall capacitance formula	-	0.5	5.000E-01

M	Area junction grading coefficient	-	0.5	2.289E-02
VJ	Junction potential	V	1.0	8.000E-01
TT	Transit-time	sec	0.0	4.121E-09

Temperature effects

Name	Parameter	Units	Default	Final value
EG	Activation energy	eV	1.11	1.110E+00
XTI	Saturation current temperature exponent	-	3.0	3.000E+00
TNOM	Parameter measurement temperature	°C	27	2.700E+01

Noise parameters and scaling factors

Name	Parameter	Units	Default	Final value
KF	Flicker noise coefficient	-	0	0.000E+00
AF	Flicker noise exponent	-	1	1.000E+00
area	Scaling factor for area	-	1	1.000E+00
perimeter	Scaling factor for perimeter	-	1	1.000E+00

ファイル名: **model_final.txt**

```

+ is=1.432E-08
+ jsw=0.000E+00
+ n=2.346E+00
+ rs=5.506E-01
+ bv=1.293E+02
+ ibv=5.000E-06
+ nbv=4.803E+01
+ ikf=1.000E-02
+ ikr=1.000E-06
+ jtun=6.750E-11
+ jtunsw=0.000E+00
+ ntun=5.238E+02
+ keg=1.000E+00
+ isr=1.639E-10
+ nr=1.489E+00
+ cjo=8.685E-13
+ fc=5.000E-01
+ m=2.289E-02
+ vj=8.000E-01
+ tt=4.121E-09
+ eg=1.110E+00
+ xti=3.000E+00
+ tnom=2.700E+01
+ kf=0.000E+00
+ af=1.000E+00
+ area=1.000E+00
+ perimeter=1.000E+00

```

5.2.2 逆方向電流-電圧特性

まずは、逆方向電流-電圧特性 (I_R - V_R) を基にモデルパラメータを合わせていきます。以下に、使用する測定回路 (Fig. 5-4) とシミュレーションスクリプト (**ir-vr.spice**) を示します。

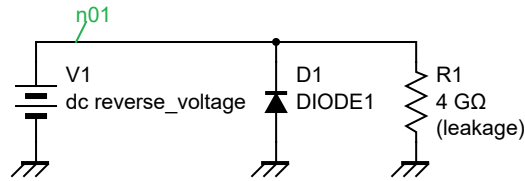


Fig. 5-4 逆方向電流-電圧特性の測定回路

ファイル名：**ir-vr.spice**

```

1  ir-vr.spice
2
3  .model DIODE1 D (
4  .include model.txt
5  + )
6
7  V1  n01 0    dc 0
8  D1  0    n01 DIODE1
9  R1  n01 0    4G
10
11 .control
12 dc V1 10 146 1
13
14 set wr_singlescale
15 wrdata ir-vr.txt -i(V1)
16 .endc
17 .end

```

パラメータ調整の流れ：

逆方向電流-電圧特性を再現するためには、各電圧領域における支配的な物理現象を意識しながら、段階的にパラメータを調整すると良いです。

- 低電圧領域（～50 V）
飽和電流パラメータ **IS** と、ダイオードに並列に挿入した高抵抗 **R1**（4 G Ω ）を調整し、ダイオードの漏れ電流に合わせます。
- 中電圧領域（70～130 V）
データシートを見ると、この領域で電流の傾きが増加する傾向が見られます。この挙動を再現するため、トンネル電流に関するパラメータ **JTUN** および **NTUN** を調整します。
- 高電圧領域（130 V 以上）
この領域では、アバランシェ降伏による急激な電流増加が支配的となります。降伏電圧 **BV** と電流の傾きを決めるパラメータ **NBVV** を調整します。**IKR** はアバランシェ降伏の立ち上がり付近（～1 μ A）に合わせます。

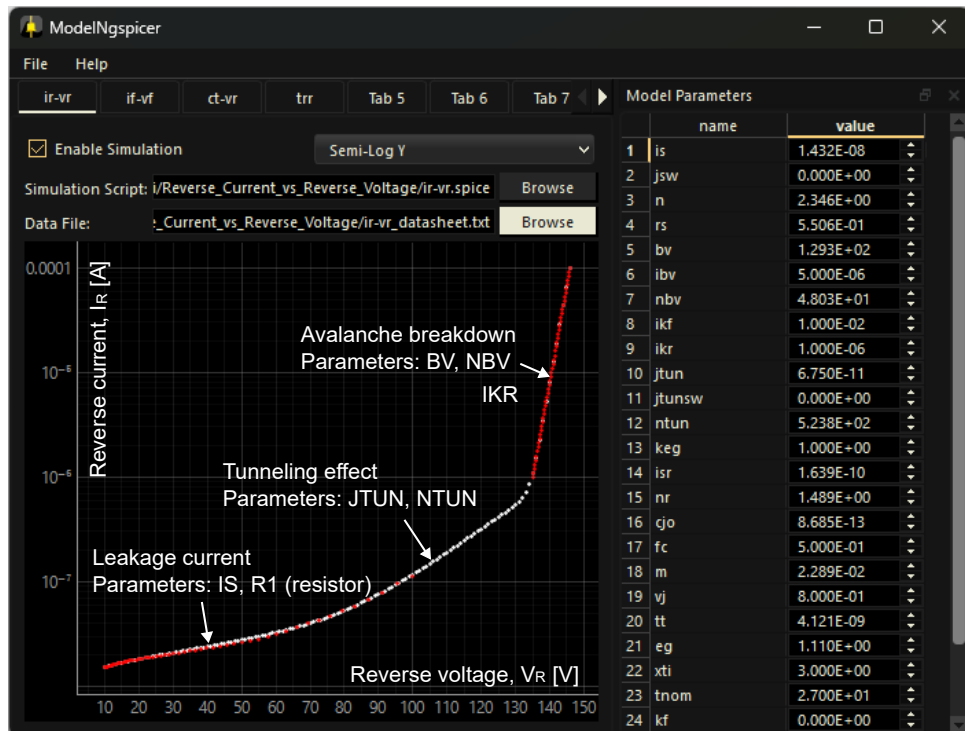


Fig. 5-5 逆方向電流-電圧特性のパラメータ調整
(赤点：データシートの読み値、白点：シミュレーション結果)

調整後の逆方向電流-電圧特性はFig. 5-5 のようになります。グラフの赤点はデータシートの読み値、白点はシミュレーション結果を示しており、両者はほぼ完全に一致しています。

5.2.3 順方向電流-電圧特性

続いて、順方向電流-電圧特性 (I_F - V_F) に基づいてモデルパラメータを合わせていきます。

以下に、使用する測定回路 (Fig. 5-6) とシミュレーションスクリプト (**if-vf.spice**) を示します。ここでは、モデルパラメータ **N**、**RS**、**IKF**、**ISR**、**NR**、**XTI** を調整することで、データシートの特性に近づけていきます。

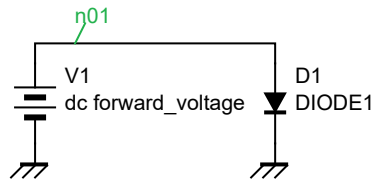


Fig. 5-6 順方向電流-電圧特性の測定回路

ファイル名：if-vf.spice

```

1 if-vf.spice
2
3 .model DIODE1 D (
4 .include model.txt
5 + )
6
7 V1  n01 0    dc 0
8 D1  n01 0    DIODE1
9
10 .control
11 foreach temp_value -40 25 65
12     option temp=$temp_value
13     dc V1 0.3 1.4 0.01
14 end
15
16 * Forward current at each temperature
17 let i1 = -dc1.i(V1)
18 let i2 = -dc2.i(V1)
19 let i3 = -dc3.i(V1)
20
21 set wr_singlescale
22 wrdata if-vf.txt i1 i2 i3
23 .endc
24 .end

```

スクリプトの解説：

11～14行目：foreachによるループ処理を用いて、 $T_a = -40^{\circ}\text{C}$ 、 25°C 、 65°C の3つの条件でDC解析を行っています。繰り返し実行したDC解析結果はdc1、dc2、dc3に順次格納されます。

16～19行目：変数i1、i2、i3には各温度における順方向電流が代入されます。

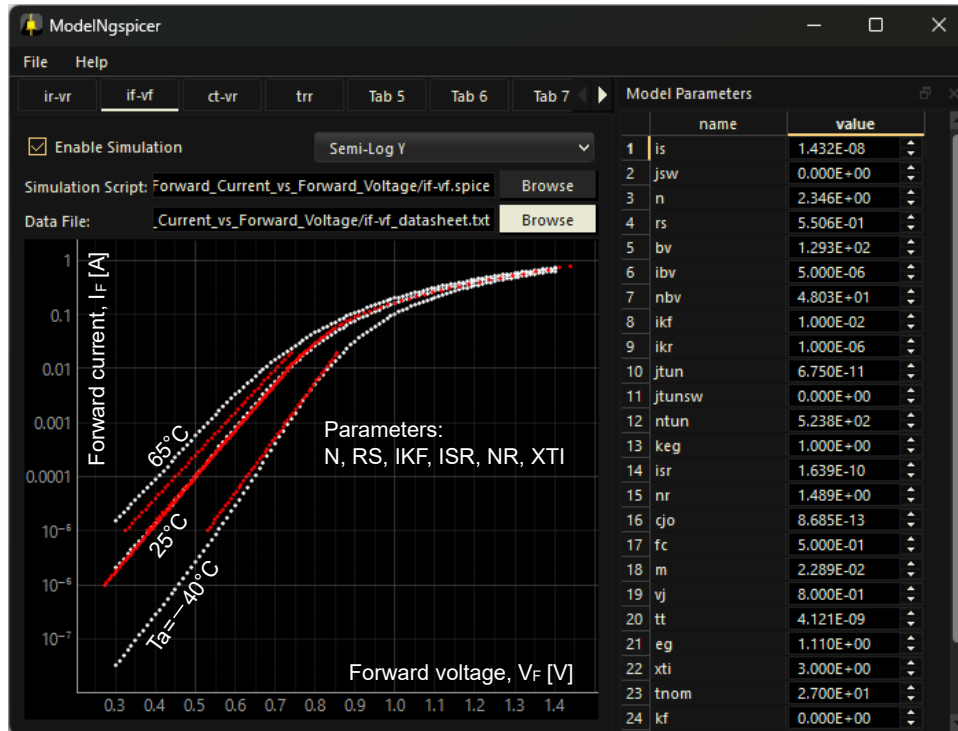


Fig. 5-7 順方向電流-電圧特性のパラメータ調整
(赤点：データシートの読み値、白点：シミュレーション結果)

調整後の順方向電流-電圧特性を Fig. 5-7 に示します。グラフの赤点はデータシートの読み値、白点はシミュレーション結果を示しており、 $T_a=25^\circ\text{C}$ の特性はよく一致しています。 $T_a=-40^\circ\text{C}$ と 65°C においては若干の乖離が見られますが、今回は許容範囲とみなし妥協することとします。

5.2.4 接合容量-逆電圧特性

次に、接合容量-逆電圧特性 (C_T - V_R) に基づいてモデルパラメータを調整します。以下に、使用する測定回路 (Fig. 5-8) とシミュレーションスクリプト (**ct-vr.spice**) を示します。ここでは、モデルパラメータ **CJO**、**M**、**VJ** を調整します。

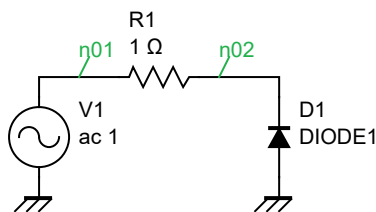


Fig. 5-8 接合容量-逆電圧特性の測定回路

ファイル名：**ct-vr.spice**

```
1 ct-vr.spice
2
3 .model DIODE1 D (
4 .include model.txt
5 + )
```

```

6
7 V1 n01 0 dc 0 ac 1
8 R1 n01 n02 1
9 D1 0 n02 DIODE1
10
11 .control
12 * Reverse voltage sweep settings
13 let st = 0
14 let sp = 15
15 let step = 0.1
16
17 let loop_index = 0
18 let loop_count = (sp-st)/step
19 let capacitance = vector(loop_count)
20 let reverse_voltage = st+step*vector(loop_count)
21
22 while loop_index lt loop_count
23     alter V1 dc reverse_voltage[loop_index]
24     ac lin 1 1Meg 1Meg
25
26     $ Calculate capacitance from admittance
27     let Y11 = v(n01,n02)/v(n02)
28     let capacitance[loop_index] = abs(imag(Y11)/(2*pi*frequency))/1e-12
29     let loop_index = loop_index + 1
30 end
31
32 setscale capacitance reverse_voltage
33 wrdata ct-vr.txt capacitance
34 .endc
35 .end

```

スクリプトの解説：

22～30 行目：while 文を用いて、電圧源 **v1** を 0 V から 15 V まで、0.1 V 刻みでスイープしています。ループ処理内で AC 解析を実行し、1 MHz におけるキャパシタンスを計算しています。計算結果は、ベクトル変数 **capacitance** に順次格納されます。

32 行目：setscale コマンドを用いて、ベクトル変数 **capacitance** と **reverse_voltage** を関連付けます。これにより、**reverse_voltage** は X 軸データ、**capacitance** は Y 軸データとして扱われます。

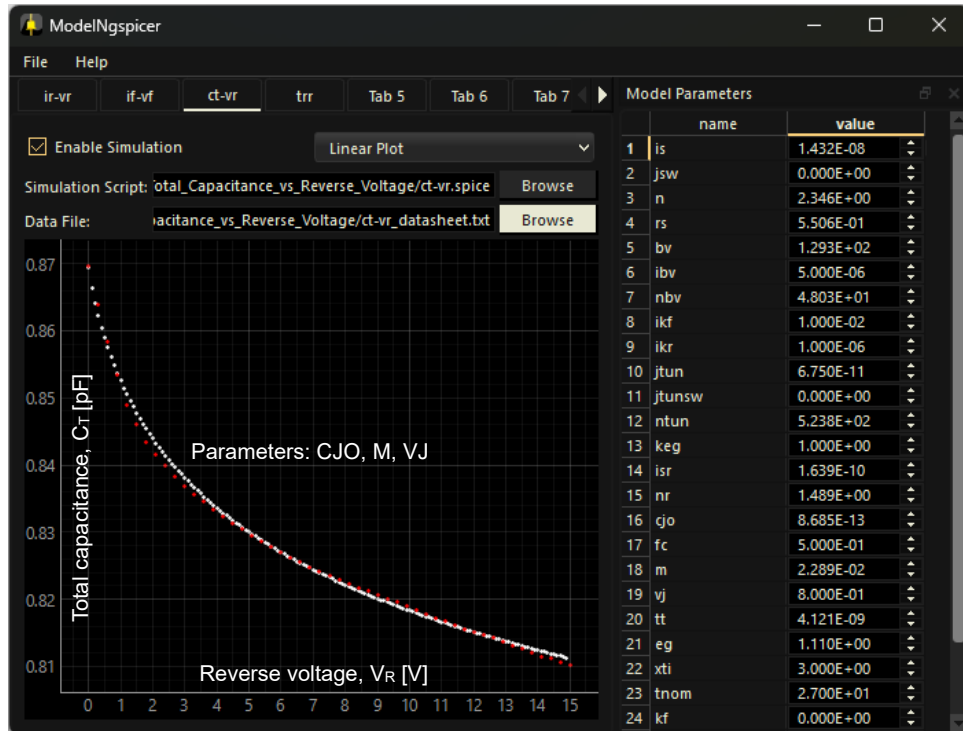


Fig. 5-9 接合容量-逆電圧特性のパラメータ調整
(赤点：データシートの読み値、白点：シミュレーション結果)

調整後の接合容量-逆電圧特性を Fig. 5-9 に示します。グラフの赤点はデータシートの読み値、白点はシミュレーション結果を表しており、両者はよく一致しています。

5.2.5 逆回復時間

最後に、ダイオードの逆回復時間 (t_{rr}) に着目して、モデルパラメータの調整を行います。逆回復時間とは、ダイオードが順方向から逆方向に切り替わる際、内部に蓄積された過剰キャリア（電子とホール）が抜けきって遮断状態になるまでの遅れ時間のことを指します。

一般的に Fig. 5-10 のような測定回路が用いられます。電圧源 **V2** より順方向電流 I_F を流した状態から、パルスジェネレータ **V1** より負のパルス印加し、逆方向電流が収束するまでの時間応答を観測します。

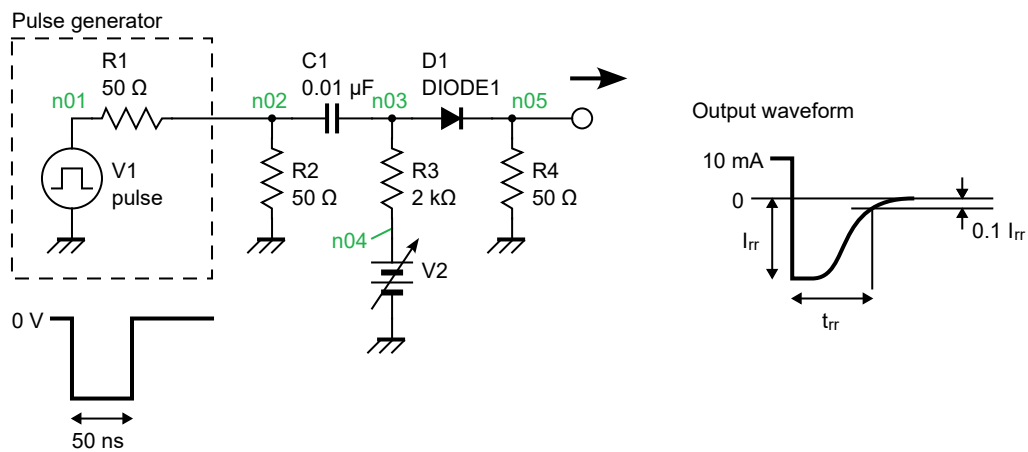


Fig. 5-10 逆回復時間の測定回路

ファイル名：trr.spice

```

1 trr.spice
2
3 .model DIODE1 D (
4 .include model.txt
5 + )
6
7 V1  n01 0    dc 0 pulse(0 -3 10n 0.1n 0.1n 50n 100n 1)
8 V2  n04 0    dc 21.2
9 R1  n01 n02 50
10 R2  n02 0    50
11 R3  n03 n04 2k
12 R4  n05 0    50
13 C1  n02 n03 0.01u
14 D1  n03 n05 DIODE1
15
16 .control
17 tran 10p 20n
18 wrdata trr.txt v(n05)
19 .endc
20 .end

```

Onsemi 社の 1N4148 のデータシートによると、 $I_F=10\text{ mA}$ 、 $I_R=10\text{ mA}$ の条件下で、逆回復時間は $t_{rr}=3.2\text{ ns}$ とされています。キャリア走行時間 (transit time) を表すパラメータ **TT** を調整することで、逆回復応答を合わせていきます。

Fig. 5-11 に調整後の逆回復応答の時間波形を示します。

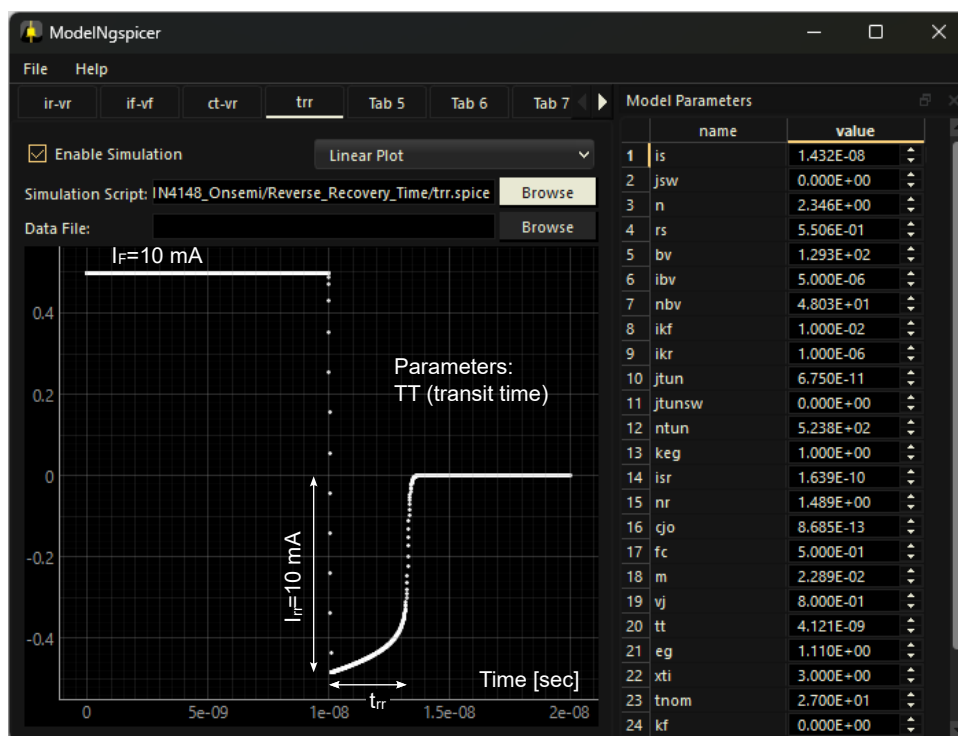


Fig. 5-11 逆回復応答のパラメータ調整