

Table Of Contents

1	INTRODUCTION	2
2	GETTING STARTED	3
2.1	RELATIONSHIP BETWEEN THE INTERPRETER AND THE SCHEMATIC WINDOW	3
2.2	CALCULATIONS	3
2.3	SIGNAL PROCESSING.....	5
2.4	SIGNAL DEFINITION	7
2.5	PROBLEM SOLVING	7
3	CALCULATIONS USING THE INTERPRETER.....	8
3.1	SOLVING DC PROBLEMS	8
3.2	SOLVING AC PROBLEMS	10
3.3	EVALUATING INTEGRALS.....	12
3.4	SOLVING LINEAR SYSTEMS	13
4	DIAGRAM DRAWING	15
4.1	DRAWING TIME DIAGRAM.....	15
4.2	BODE DIAGRAMS.....	17
4.3	DRAWING IMPEDANCES.....	20
5	CALCULATIONS AND DIAGRAM DRAWING.....	22
5.1	AN EXAMPLE OF CONVOLUTION	22
5.2	SOLUTION	22
5.3	VERIFICATION	24
6	PROCESSING TINA'S RESULTS.....	25
6.1	CALCULATING AVERAGE POWER	25
7	SIGNAL DEFINITION.....	28
8	USING THE INTERPRETER IN THE TRAINING AND EXAMINATION MODE OF TINA	29
8.1	SOLVING A DC PROBLEM.....	29
8.1.1	<i>Using the Interpreter in the Result field</i>	<i>29</i>
8.1.2	<i>Submitting a solution from the Interpreter window.....</i>	<i>30</i>
8.2	SOLVING AN AC PROBLEM.....	31
9	APPENDIX.....	32
10	INDEX.....	35

The Interpreter

1 Introduction

The Interpreter is a powerful tool which extends TINA's usefulness. It allows the evaluation of mathematical expressions, the solution of equations and systems of equations, the calculation of derivatives and integrals, the plotting of results, and much more.

You may liken the Interpreter to using paper and pencil for solving problems, except that you can have TINA check the results. If you use the Interpreter after DC, AC or Transient analysis, it is extremely handy for post-processing the results.

Another interesting application of the Interpreter is the arbitrary definition of signals to be used as excitation for a circuit. These signals can be assigned to analog generators.

For educational users, TINA has two special modes. In examination mode, the student has to solve a series of problems (*a problem set*) either by traditional pencil-and-paper methods or by using the *Interpreter and analysis functions*. When the student finds the answer, the program sends it immediately to the teacher's machine, where it is promptly displayed by the TSuper supervision utility. In training mode, operation is similar, except that TINA gives the student feedback about the correctness of his answer, and the student may turn to the Advisor to get help prepared by the teacher.

The Interpreter can be activated from the Tools menu. Its interface is similar to that of TINA's text editor: you enter expressions as text and then execute it as a program or an expression to evaluate or plot.

2 Getting Started

In this chapter we will consider some simple examples, to see the types of problems we can solve with the Interpreter.

2.1 Relationship between the Interpreter and the Schematic Window

Although the Interpreter can be used entirely on its own with no schematic file open in the Schematic Window, it gains in power when used in relation to an underlying schematic. When a schematic file is open in the Schematic Window, you can access its circuit variables and component parameters.

Note that, in order to make component parameters accessible, you must enter a component reference symbol in the Label field of each component's properties window. The label for a 250 ohm resistor, for example, could read *R,250*, *R*, or *R=250*. With any of these three labels, the Interpreter will use *R* as the reference name and will pick up the value from the Value field. The labels showing the value aid in user documentation on the schematic. In the case of components such as inductors or capacitors, additional parameters (parallel resistance or series resistance, respectively) become available. In general, each component reference symbol should be unique (an exception being where there are multiple components all with the same values).

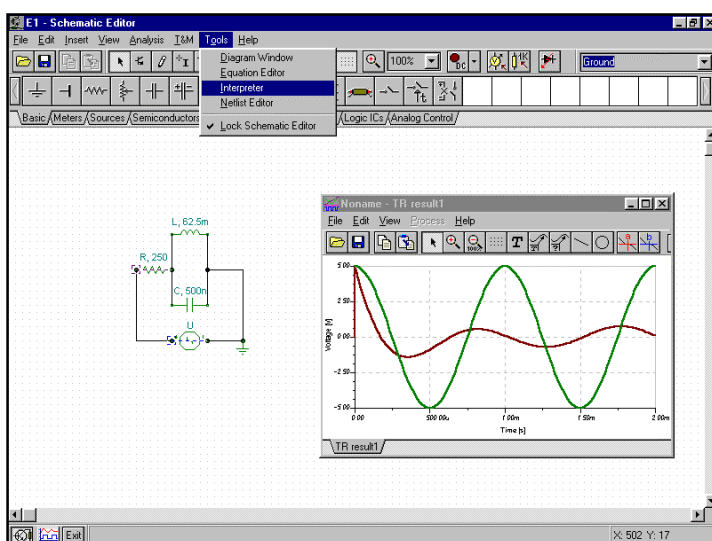
The various parameters of sources (e.g. for a sine wave, DC level, AC amplitude, frequency, and phase) in a schematic are also available to you in the Interpreter. They appear in the symbol table under the heading Circuit variables. Be sure to fill in the label field for sources.

Finally, the results of analyses run on the schematic are also available as functions (of time or frequency). These appear in the Interpreter symbol table as External Functions, under the names they have been assigned by the Diagram Window (e.g. *AC_Ampli2*, *AC_Group_delay3*).

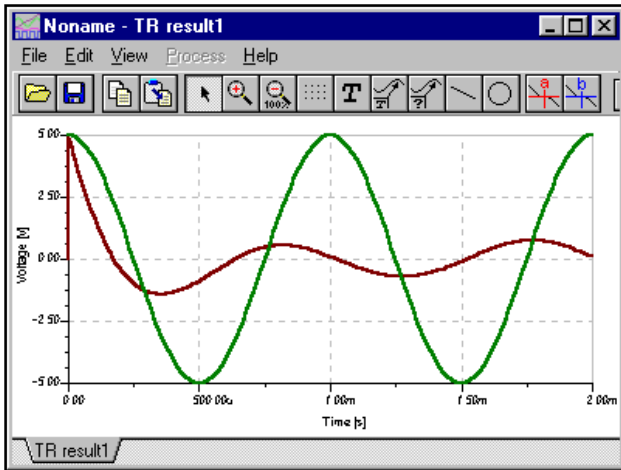
2.2 Calculations

In the following example we will see how to do simple calculations in the Interpreter. The Schematic Editor will contain a basic RLC circuit. (*e1.sch*)

Run transient analysis with the given defaults. Click on the Tools menu and select the Interpreter. The Interpreter window appears.



Note: To insert these TINA screen views into this manual (a Microsoft Word document), we used the *PrintScreen* key in TINA to copy the screen to the clipboard. You could also copy any windows of TINA to the clipboard by selecting the window and pressing the *Alt+PrintScreen* keys. Then we pasted the clipboard contents into this document.



i) Let's first check R , one of the circuit parameters.

Write in the Interpreter

$R =$ *Then press enter.*

The Interpreter will give the result

$R = [250]$ *That is 250 Ohm*

ii) Calculate the impedance of the circuit at a radian frequency of 100k:

$$W(s) = R + sL \parallel \frac{1}{sC}$$

Write in the Interpreter,

$R = [250]$

$w := 100k$

$Z := R + \text{Replus}(j * w * L, 1 / (j * w * C))$

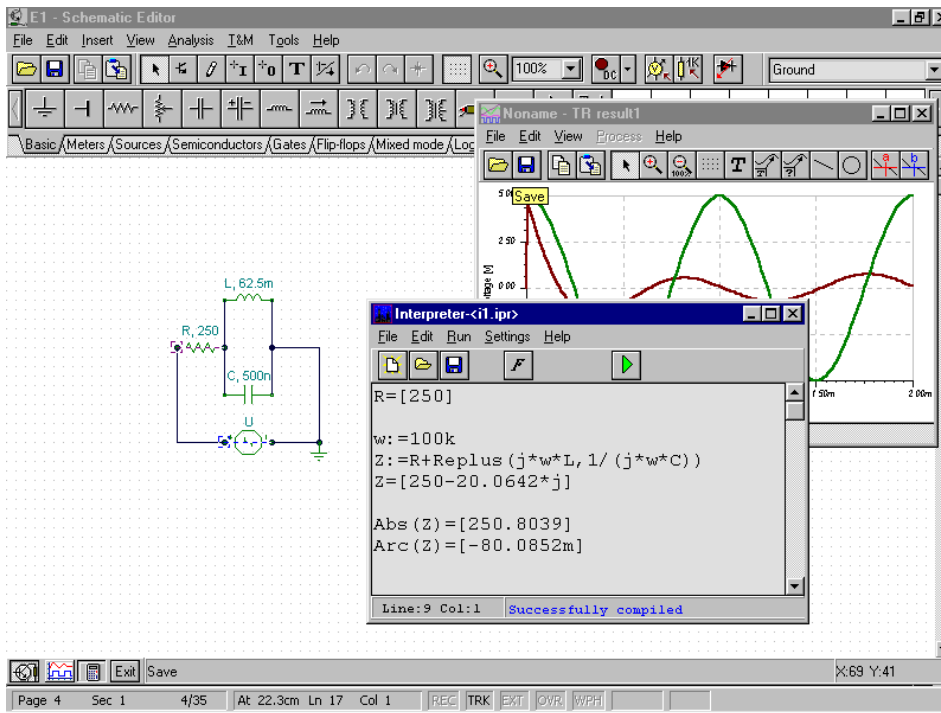
$Z = [250 - 20.0642 * j]$

$\text{Abs}(Z) = [250.8039]$

$\text{Arc}(Z) = [-80.0852m]$

{ the absolute value }

{ the phase value(radian) }



There are more functions available (See the *Appendix*).

To save these calculations, click the *Save* button on the toolbar, and follow the prompts. To recalculate results, press the *Run* button on the toolbar or use the Run menu.

2.3 Signal processing

We can use TINA to post-process the results of an analysis of a circuit. After a DC, AC or transient analysis has been performed on a schematic, the analysis results are placed in the symbol table of the Interpreter so you that can refer to them as you write expressions in the Interpreter.

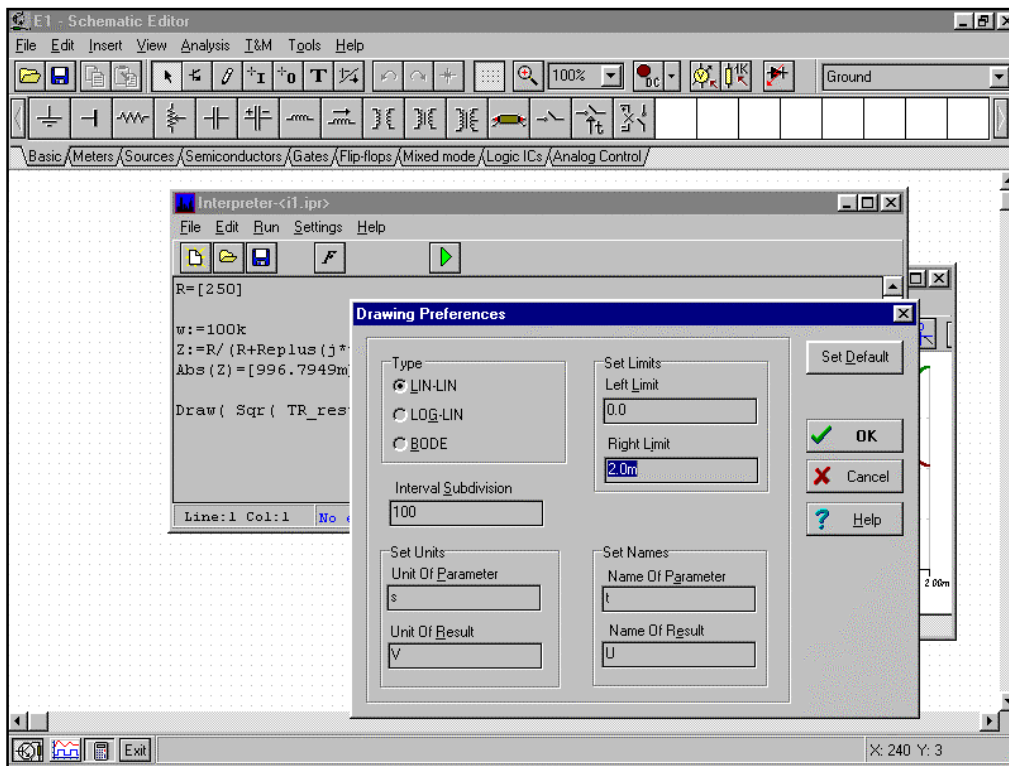
In this example, we will plot the instantaneous power in the resistor using the previous TINA transient analysis result.

The instantaneous power in the resistor is:

$$p(t) = u(t) \cdot i(t) = \frac{u^2(t)}{R}$$

THE INTERPRETER

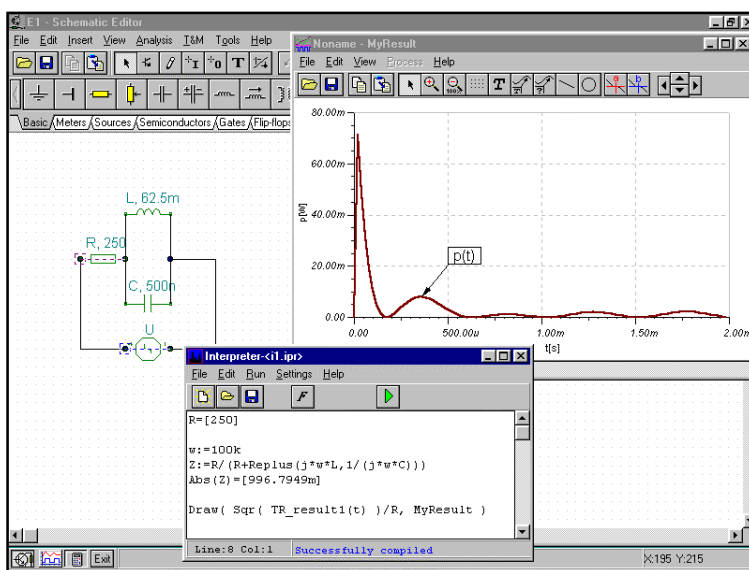
Before plotting this signal, we have to set the ending interval time value (right limit parameter) in the *Settings | Drawing Preferences* dialog. Set this parameter to the value of the transient analysis end time, 2.0m.



Type in the Interpreter:


```
Draw( Sqr( TR_result1(t) )/R, MyResult )
```

The result appears after a moment in the Diagram Window.



2.4 Signal definition

Another interesting application of the Interpreter is the arbitrary definition of signals to be used as excitation for a circuit. These signals can be assigned to analog generators. User defined waveforms are extremely general and call up an equation editor window where the waveform may be defined as a mathematical expression. To make signal definition even easier, a template which you can modify to create your own signal is provided in the editor window. *Note that while TINA's Interpreter is utilized for signal definition, you access signal definition via the Schematic Editor window, not the Tools/Interpreter menu.*

As a simple example of this feature, pick up a voltage generator from the toolbar, place it in the *Schematic Editor* window, and then double-click on it. The dialog box of the Voltage Generator will appear. Click on Unit Step in the Signal field and then press the  button that appears. A new dialog box opens showing the available signals. Now press the rightmost button, *User Defined*, in the *Signal Editor* window. The curve and the description of a simple linear time function will appear. To change this into a quadratic function, replace t with $t*t$ in the description on the right side and then press the *Test* button. The waveform versus time diagram of the new function will appear. Now you can use this new signal with any circuit during transient analysis. For more details of User defined excitations, refer to *Chapter 7*.

2.5 Problem Solving

TINA has two special modes for educational purposes. In examination mode, the student has to solve a series of problems (*a problem set*), either using pencil-and-paper or the *Interpreter* and analysis functions. When the student finds the answer, the program sends it immediately to the teacher's machine, where it is promptly displayed by the TSuper supervision utility. Operation is similar in training mode, except that TINA gives the student feedback about the correctness of his answer. Furthermore, in training mode the student may turn to the Advisor to get help prepared by the teacher.

To learn more about this feature see *Chapter 8*.

3 Calculations using the Interpreter

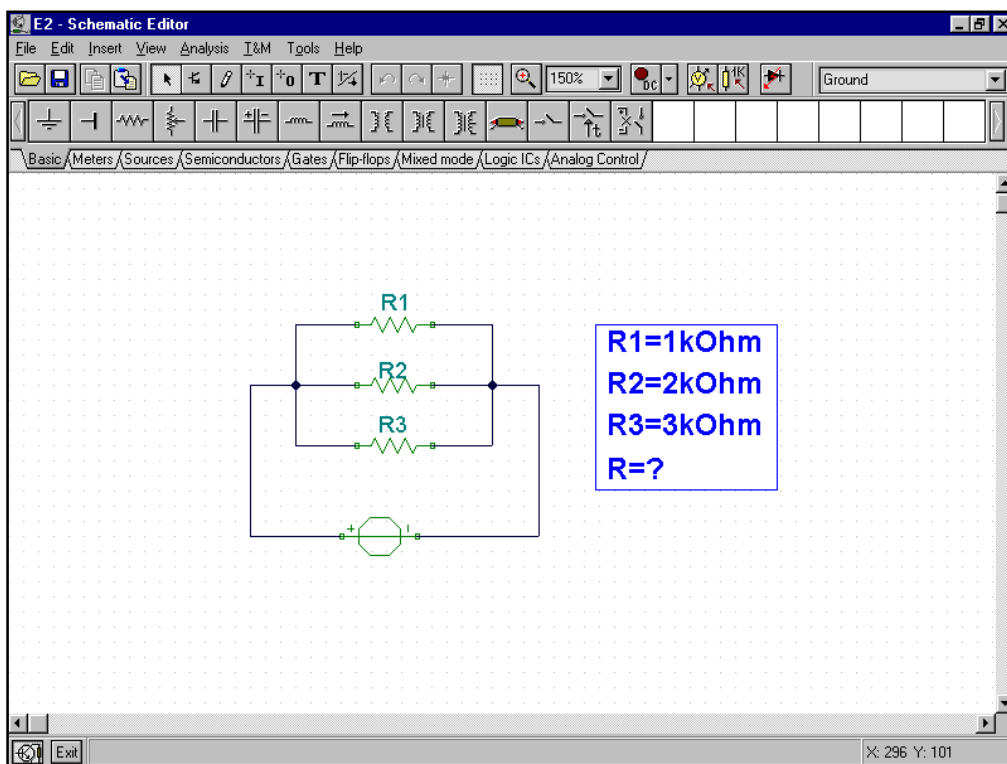
You can also use the Interpreter to carry out calculations. In this chapter, we will present examples of the calculations that you can do with the Interpreter.

To solve a problem, you first create the “source code” that the Interpreter will compile and execute. This code is structured in two parts: the first part contains function definitions, while the second part contains executable commands. Functions help you to solve complicated problems.

3.1 Solving DC problems

Example

Resistors R1, R2, R3 in the circuit below are connected in parallel. Calculate the total resistance.



Solution

i) We will solve the problem first by entering the equation for R. Write in the Interpreter

```
R := 1 / ( 1/R1+1/R2+1/R3 )
R =, Then press enter
```

The Interpreter will give the result,

```
R = [545.4545]
```


THE INTERPRETER

Since we have a schematic file open, we can access variables derived from the component name labels--R1, R2, and R3, for example. Click on the menu to *Settings/View Symbol Table* to view which variables are available to you.

ii) We will define a function to compute the total resistance of arbitrary parallel connected resistances labeled R1, R2, R3. The Interpreter's Function declaration is used to solve more complicated problems. The syntax of the function declaration is:

```
Function FunctionName(parname1,...,parnamen);  
Begin  
    function body  
End;
```

The function that computes the parallel resistance is:

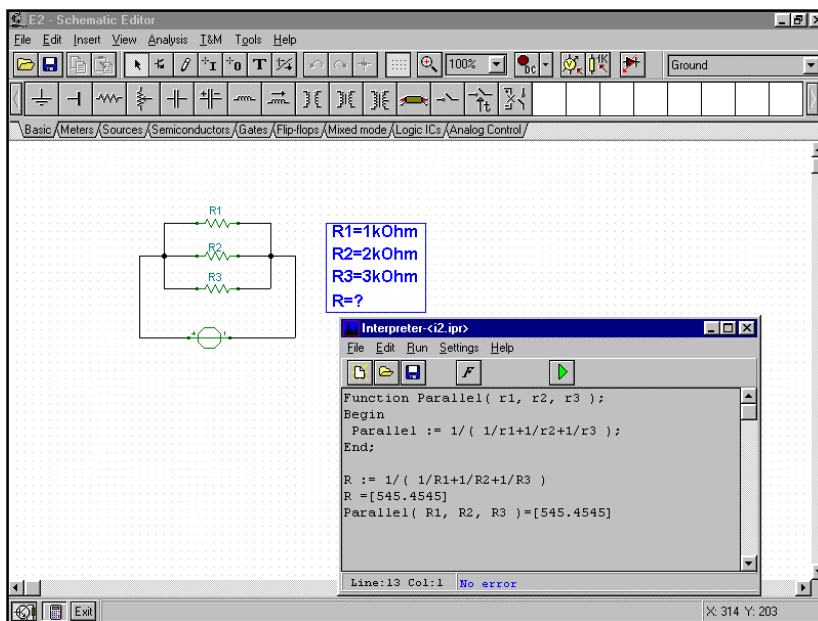
```
Function Parallel( r1, r2, r3 );  
Begin  
    Parallel := 1/( 1/r1+1/r2+1/r3 );  
End;
```

For a numerical solution based on the values of R1, R2, and R3 in the underlying schematic, type the following and press Enter:

```
Parallel( R1, R2, R3 )
```

See the result:

```
= [545.4545]
```

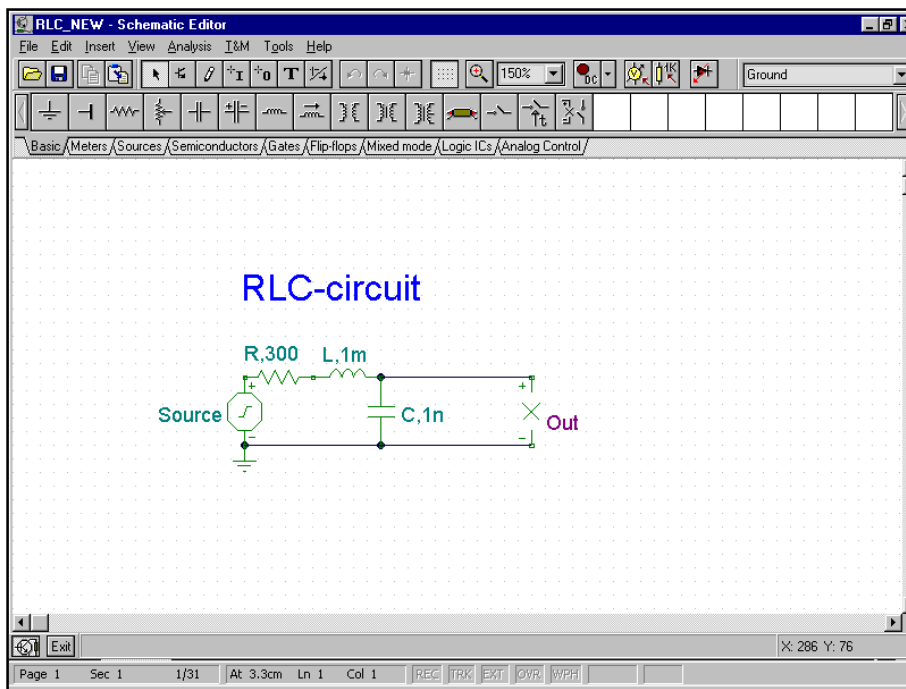


3.2 Solving AC problems

Example

Consider the following circuit. Calculate the phase and absolute value and the time function of the voltage named *Out*, when $f=1\text{MHz}$. The time function of the voltage generator is:

$$V_{source}(t) = 10 \cdot \cos(\omega \cdot t + 45^\circ)$$



Solution

First we find the AC transfer function,

$$W(s) = \frac{\frac{1}{s \cdot C}}{R + s \cdot L + \frac{1}{s \cdot C}} = \frac{1}{1 + s \cdot R \cdot C + s^2 \cdot C \cdot L}$$

Then we represent the complex source and its peak value,

$$V_{source} = 10 \cdot e^{j \cdot 45^\circ}$$

The solution will be,

$$V_{out}(j \cdot \omega_0) = V_{source} \cdot W(j \cdot \omega_0)$$

where ω_0 is the desired frequency.

In our program, we first define a variable **s** to hold the complex frequency value, $s=j\omega_0$. Then we calculate the transfer function $W(s)$ using the complex peak value V_{source} . By default, TINA gives us the phase angle in radians. To present it in degrees, simply change the *Settings/Numeric formats & precisions/Angle* to DEG

With the help of the Interpreter,

```
s := j*2*pi*1e6
W := 1/(1+R*C*s+C*L*s^2)
VSource := 10*exp(j*pi/4)    {Complex peak value of the source}
VOut := VSource*W
VAbs := Abs( VOut )          {Abs. value of the output voltage }
VFi := Arc( VOut )           {Phase value of the output voltage}
VFi=[3.9759]
VFiDeg := RadToDeg( VFi )

VAbs=[259.5747m]
VFiDeg=[227.8045]
```

Using these results, we write the output voltage:

$$V_{out}(t) = V_{Abs} \cdot \cos(\omega \cdot t + V_{Fi}) = 0.259 \cdot \cos(\omega \cdot t + 3.97)$$

3.3 Evaluating integrals

Example

Another application of the Interpreter is numerical solution of arbitrary integrals.

Calculate $\int_{-\pi}^{+\pi} (\sin 2x + \cos^2 x) dx$

Solution

We define a function $f(x)$ to hold the function to be integrated; then we compute the integral using TINA's *Integ* function.

```
Function f(x);  
Begin  
  f := sin(2*x)+Sqr(cos(x));  
End;  
Integ( f(s), -pi, pi, s )=[3.1416]
```

The syntax of the *Integ* command used above is:

Integ(Expression(τ),Limit1,Limit2, τ)

Expression(τ) is the expression to be integrated; the left and right limits are Limit1, Limit2; and the variable of integration is τ .

There are a large number of built-in functions that you can use in the Interpreter. The formal parameters of some functions (such as *sin*, *cos*, *exp*, etc.) are complex. See the *Appendix* for details.

3.4 Solving linear systems

One of the most powerful features of the Interpreter is the linear equation solver. With it, there is no need to reduce the equations manually to achieve a solution. The syntax of the equation solver is:

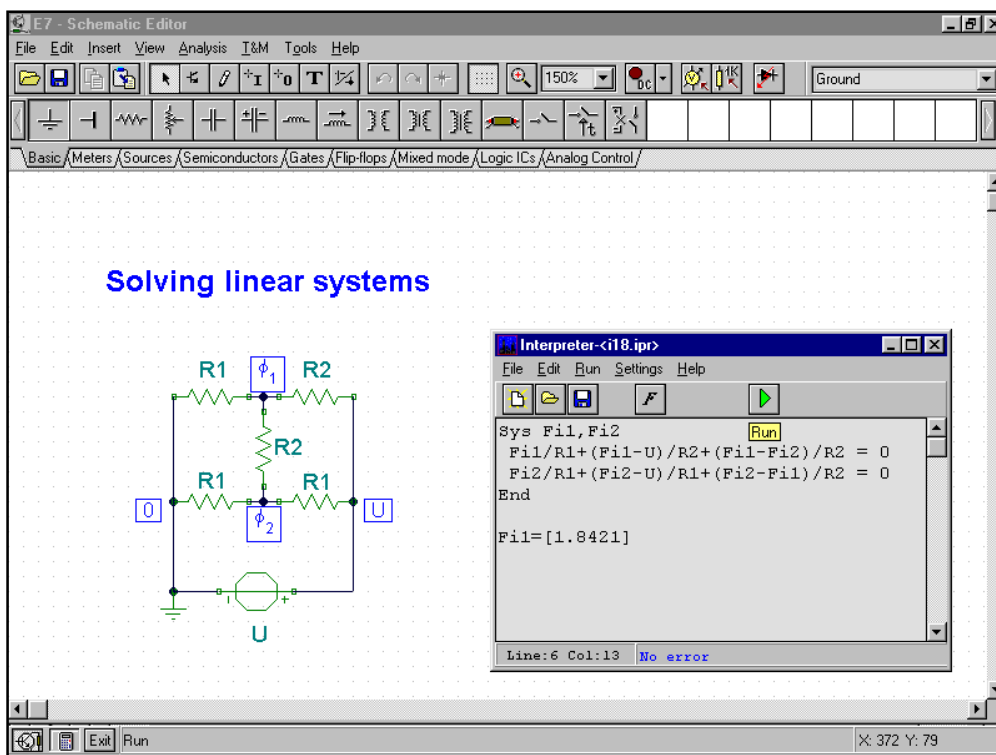
```
sys name1,...,namen
  equation1
  .
  .
  equationn
end;
```

where equation i is the i th equation, and name i is the i th unknown variable. If there is no solution, or if there are multiple solutions, the Interpreter will present an error message.

Our linear system example will be the solution of a system of nodal equations.

Example

Let's consider the following circuit. Determine the nodal voltage at Φ_1 .



Solution

We need only enter the correct equations: the Interpreter will then solve them automatically. The nodal equations are:

$$\frac{\phi_1}{R_1} + \frac{\phi_1 - U}{R_2} + \frac{\phi_1 - \phi_2}{R_2} = 0$$

$$\frac{\phi_2}{R_1} + \frac{\phi_2 - U}{R_1} + \frac{\phi_2 - \phi_1}{R_2} = 0$$

Note that we assume that the underlying schematic as shown, with component values, has already been created. We will reference variables from the schematic.

Entering these as source code in the Interpreter:

```
Sys Fi1,Fi2
  Fi1/R1+(Fi1-U)/R2+(Fi1-Fi2)/R2 = 0
  Fi2/R1+(Fi2-U)/R1+(Fi2-Fi1)/R2 = 0
End

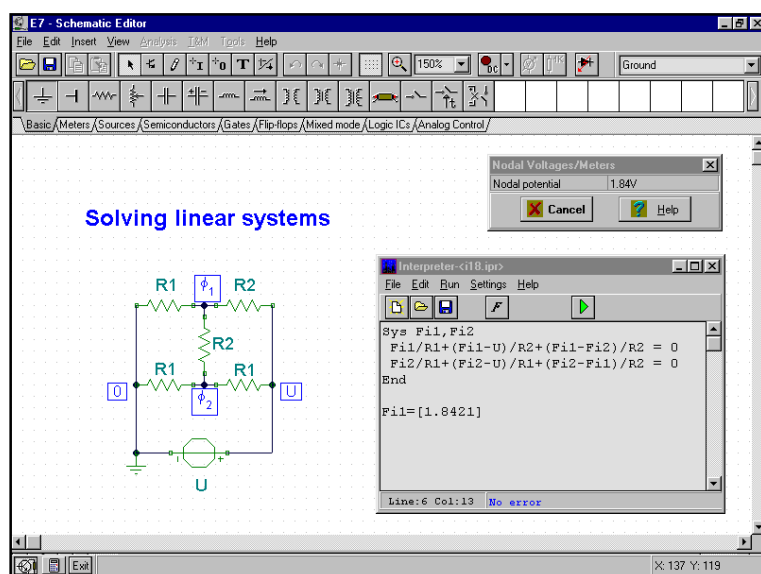
Fi1=[1.8421]
```

In this example, the system of equations of variables $Fi1$, $Fi2$ is solved assuming that U , $R1$, $R2$ are already defined. In TINA, these parameters are automatically referenced by the labels of the appropriate components. To help in documentation, we used TINA's text editor to put the Φ_1, Φ_2 node variable labels on the schematic as text items.

Verification

We use the *Analysis/DC Analysis/Calculate nodal voltages* to check the solution obtained by the Interpreter. After choosing *Calculate nodal voltages*, the cursor changes into a pencil shape. Click with the cursor on the Φ_1 node to verify its voltage.

Using *Calculate nodal voltages*, we get $1.84V$, which agrees with the value computed by the Interpreter.



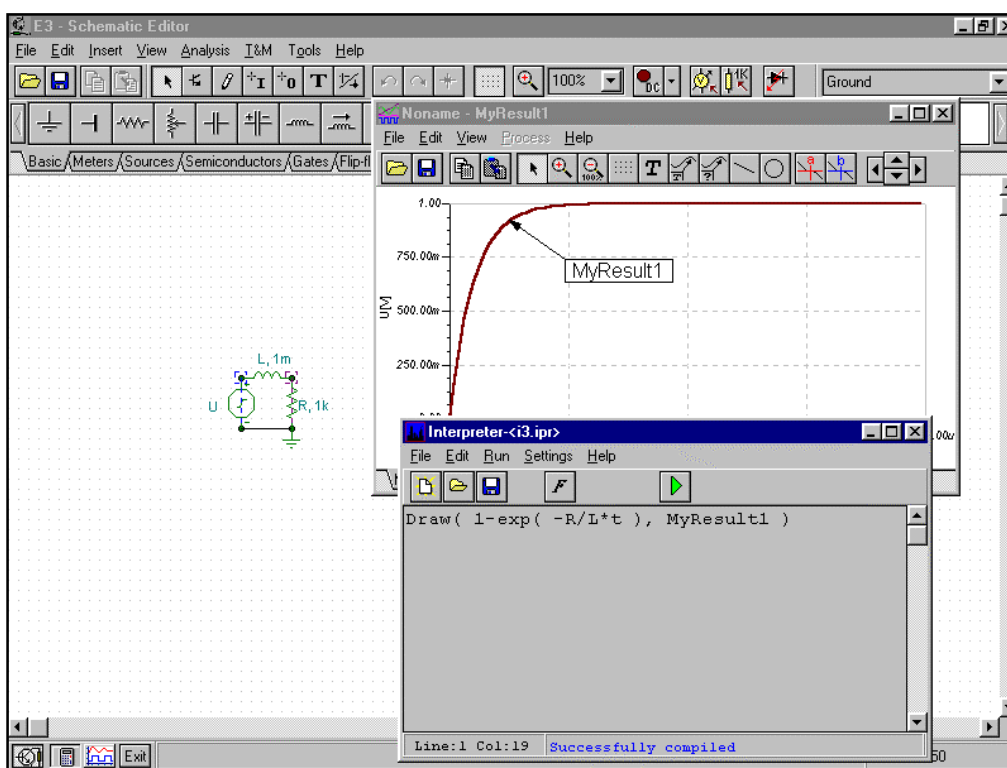
4 Diagram drawing

When you use the Interpreter you can access the results of previous analyses. You can also define and plot arbitrary functions and compare the theoretical solution with the analysis result.

4.1 Drawing time diagrams

Using the Interpreter, you can define arbitrary functions of time and plot the results in the time domain.

Referring to the following RL circuit entered as a schematic diagram, run transient analysis with the given defaults. Then use the Interpreter to plot the theoretical result.



i) The time response is:

$$u(t) = 1 - e^{-\frac{R}{L}t}$$

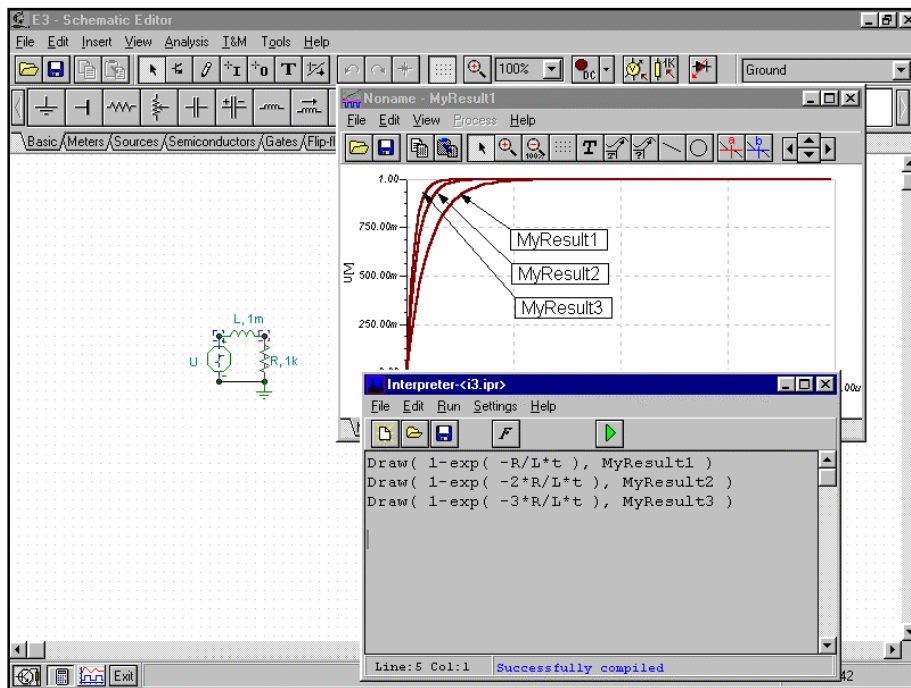
Enter this response directly in the Interpreter, referring to the schematic parameters and using the Draw command:

```
Draw( 1-exp( -R/L*t ), MyResult )
```

We can compare the computed result and the theoretical result (in this case, the results are almost the same).

ii) Changing the R parameter, we can see how the output signal changes. We use three instances of the Draw command, each with a different R parameter, to place three curves on the diagram:

```
Draw( 1-exp( -R/L*t ), MyResult1 )
Draw( 1-exp( -2*R/L*t ), MyResult2 )
Draw( 1-exp( -3*R/L*t ), MyResult3 )
```

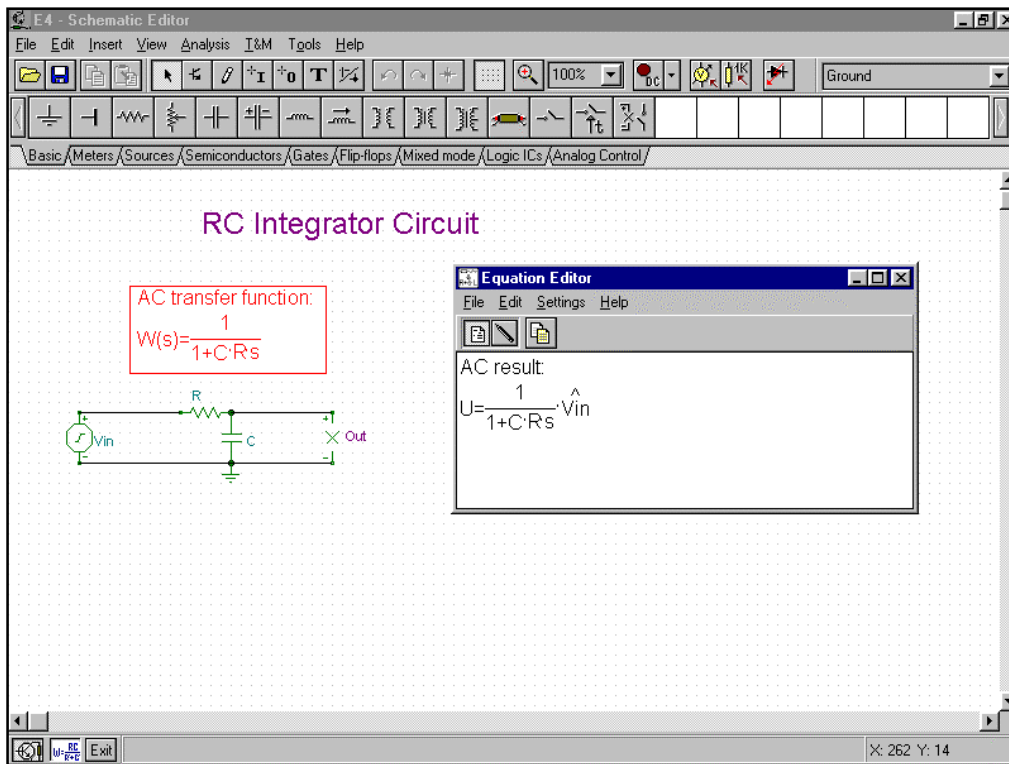


4.2 Bode diagrams

You can define arbitrary functions in the frequency domain and plot them.

Consider the following RC circuit. Draw the *AC transfer function* and calculate the amplitude and phase for $f = 5\text{kHz}$.

Run AC analysis with the given defaults. Using Symbolic Analysis we can get the AC transfer function.



Drawing the AC transfer function.

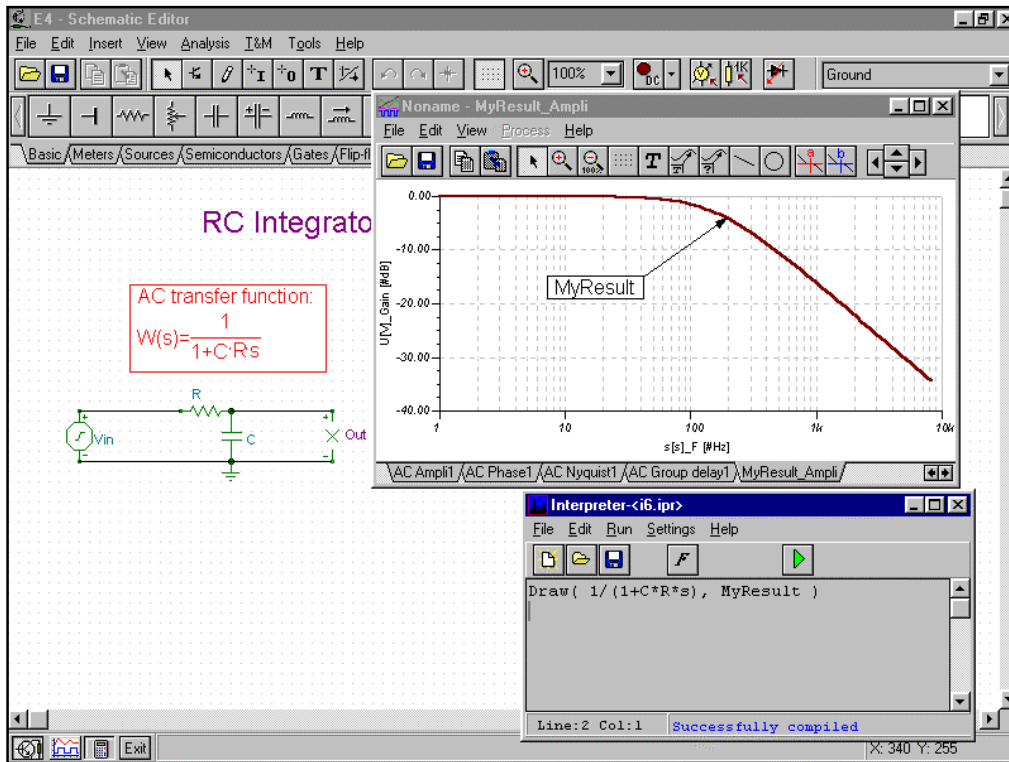
Now let's use the Interpreter to draw the AC transfer function with the given defaults. Before plotting the result, we have to set some parameters in the Drawing Preferences dialog.

- Set the *left and the right limit* parameter in the Drawing Preferences dialog. Set these to the AC analysis *start and end frequency*.
- Set the *name of parameter* to s in this dialog box also.
- Set the *type* to Bode.

Then we can plot the desired function:

```
Draw( 1/(1+C*R*s) , MyResult )
```

Two new pages will be opened in the Diagram Window, for the amplitude and the phase diagrams of the function.



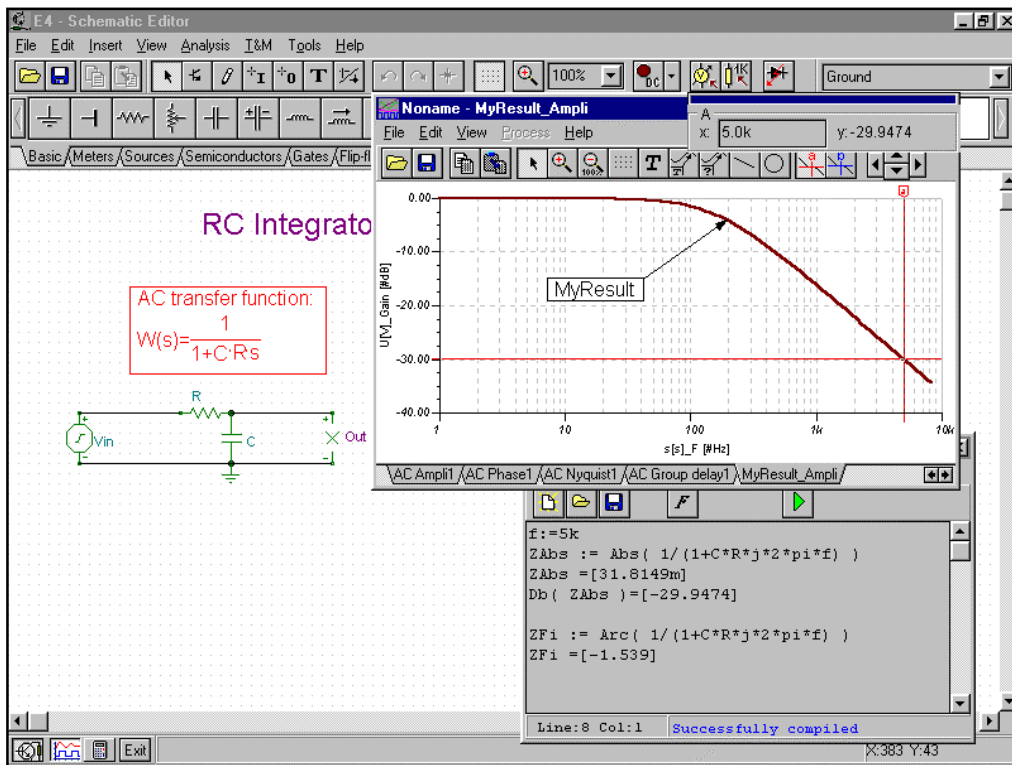
Calculating the amplitude and the phase when $f = 5\text{kHz}$

The Interpreter provides a number of built-in functions that help in solving complicated problems. Using the *Abs* and the *Arc* functions, we can solve this problem easily.

```
f := 5k
ZAbs := Abs( 1/(1+C*R*j*2*pi*f) )
ZAbs = [31.8149m]
Db( ZAbs ) = [-29.9474]

ZF1 := Arc( 1/(1+C*R*j*2*pi*f) )
ZF1 = [-1.539]
```

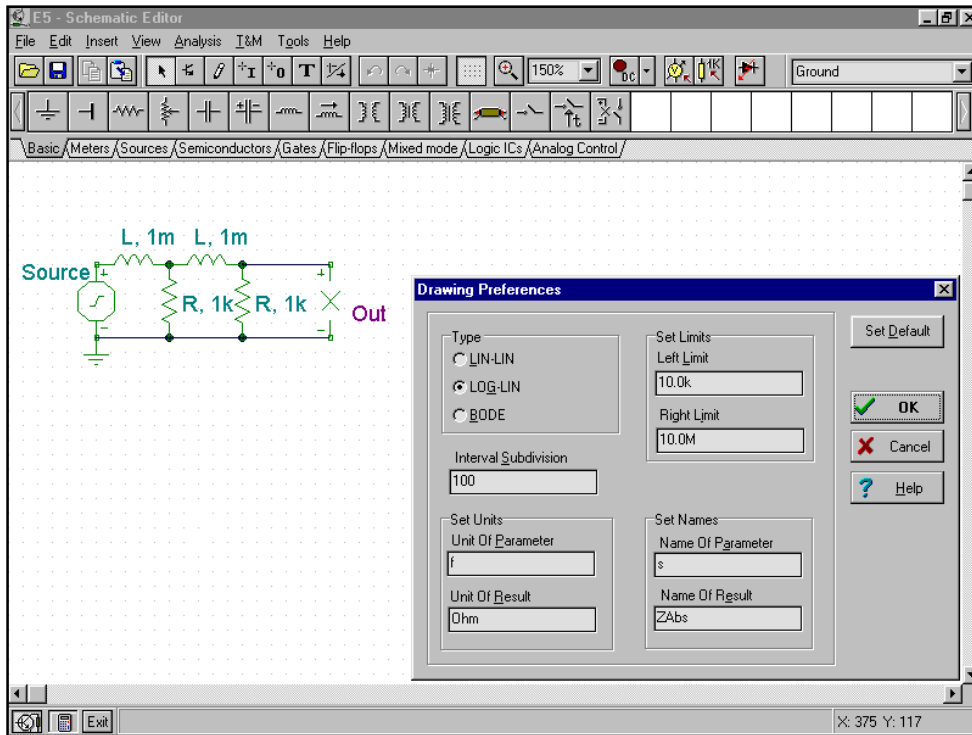
In this example, we define a variable f for frequency and set it to the given value. Then, using the *Abs* function, we calculate the absolute value (magnitude) of the transfer function at the given frequency. Next, using the *Db* function (returns $20 \cdot \log_{10}(x)$, provided $x > 0$), we get the amplitude in decibels. We can check these results by running an AC Analysis on the circuit in the schematic window, going to the diagram window, and placing a cursor over the amplitude or phase curves at 5 kHz. Finally, we use the *Arc* function to get the phase angle in radians per second.



4.3 Drawing impedances

The Interpreter can also draw impedance functions, with a choice of *linear*, *log-lin* or *Bode* diagrams.

Starting with the RL circuit shown below, we will use the Interpreter's LOG-LIN diagram to draw the impedance as a function of frequency. We are after the impedance that the source sees looking into the network from its terminals. Note that we have two identically named and valued inductors and two identically named and valued resistors. If the resistors had different values, we would have to label them with different reference designators in order for the Interpreter to pick up the correct values.



Drawing the impedance in the frequency domain

Before plotting the impedance function, we have to setup some parameters in the Interpreter's Drawing Preferences dialog.

- Set the *left and the right limit* parameter in the Drawing Preferences dialog. Set this parameter to the AC analysis *start and end frequency*. (10k and 10M)
- Set the *name of parameter* to *s*.
- Set the *name of result* to *ZAbs*.
- Set the *unit of parameter* to *f*
- Set the *unit of result* to *Ohm*
- Set the *type* to LOG-LIN.

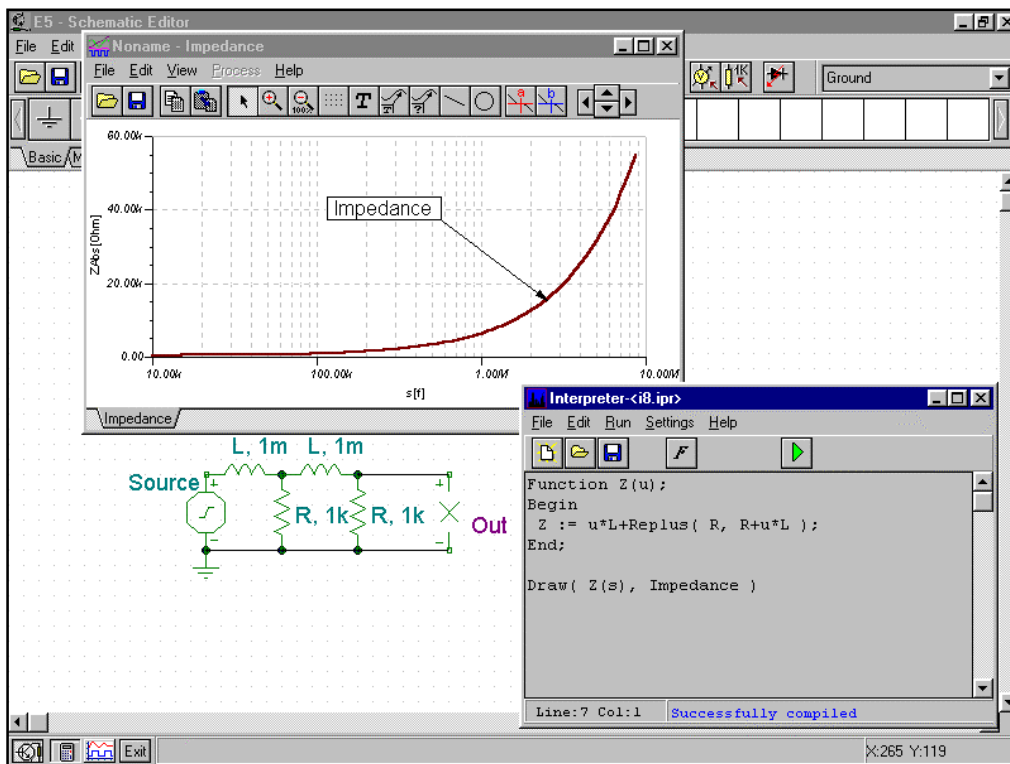
We can draw the desired function:

```
Function Z(u);
Begin
  Z := u*L+Replus( R, R+u*L );
End;

Draw( Z(s), Impedance )
```

At first we define a function to hold the impedance expression. Note in the function that u is a parameter that is replaced with s when the function is actually “called” in the Draw statement. We use here the built-in *Replus* function to calculate the impedance of two parallel connected impedances. The *Replus* function accepts either complex or real arguments. Then, using this function in the Draw statement, and relying on the current component values of the underlying schematic, we draw the impedance function in the Diagram Window.

Two new pages are created in the Diagram Window--the amplitude and the phase diagrams of the given function.

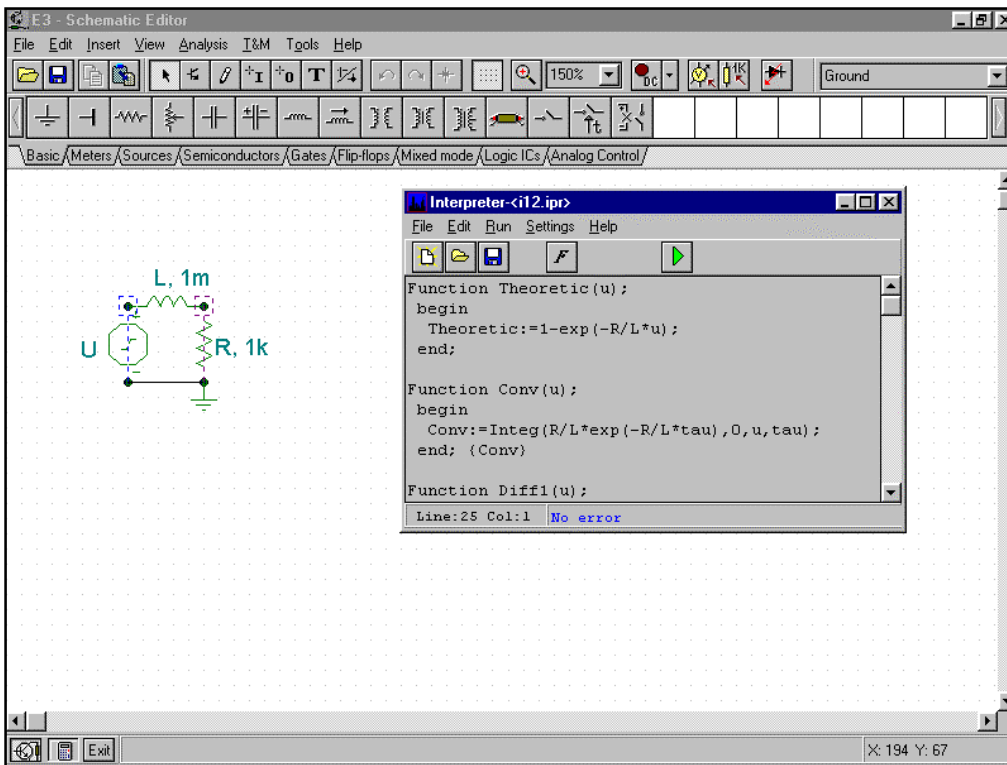


5 Calculations and diagram drawing

In this chapter we will use convolution to obtain the response of a circuit in the time domain. Convolution is a mathematical procedure in circuit theory that is used to determine the response in time domain.

5.1 An example of convolution

Let's use convolution to find the time domain response of the RL circuit below. The input to the circuit is a *unit step* voltage generator and the output is the output voltage on the resistor R.



5.2 Solution

The theoretical response of this circuit in the time domain is,

$$U_R(t) = 1 - e^{-\frac{R}{L}t}$$

What does convolution mean and how can it be used to determine the response of this circuit in the time domain? The convolution of two arbitrary signals is:

$$f_1(t) * f_2(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) d\tau$$

It can be proven that the time response of a circuit to a source $s(t)$ is:

$$y(t) = w(t) * s(t) = \int_{-\infty}^{\infty} w(\tau) s(t - \tau) d\tau$$

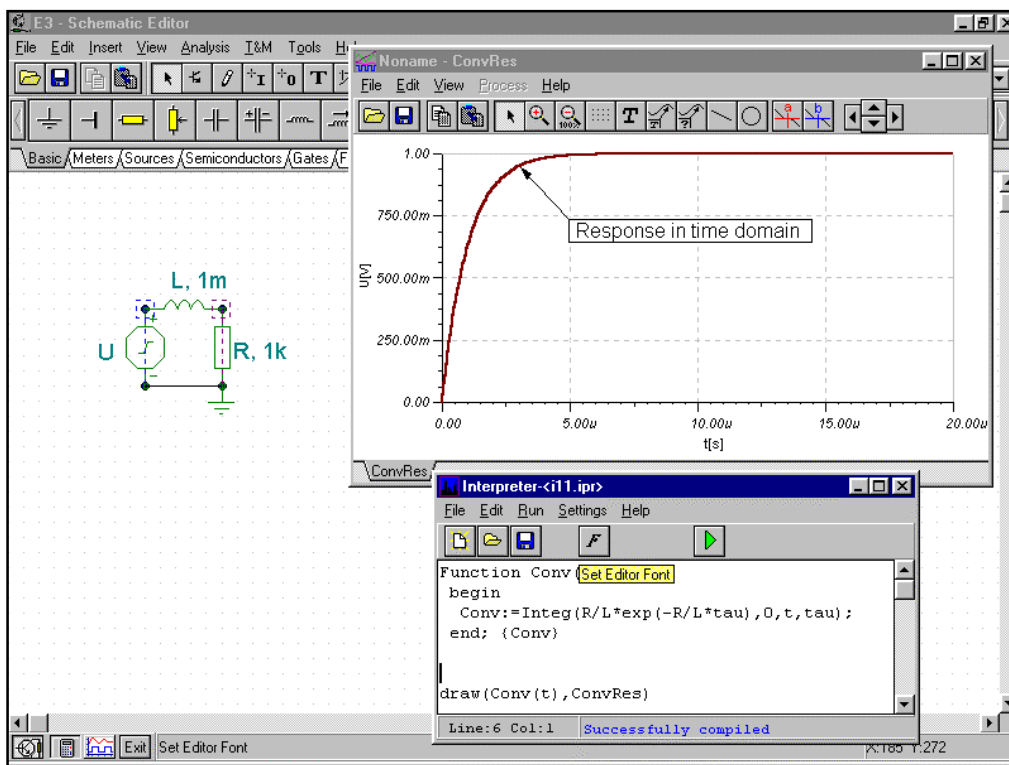
where $w(t)$ is the response of the circuit to the Dirac impulse. In this case,

$$w(t) = \frac{R}{L} e^{-\frac{R}{L}t}$$

So, we have to define a function that computes the desired convolution integral.

```
Function Conv(u);
begin
  Conv:=Integ(R/L*exp(-R/L*tau),0,u,tau);
end; {Conv}

draw(Conv(t),Conv)
```



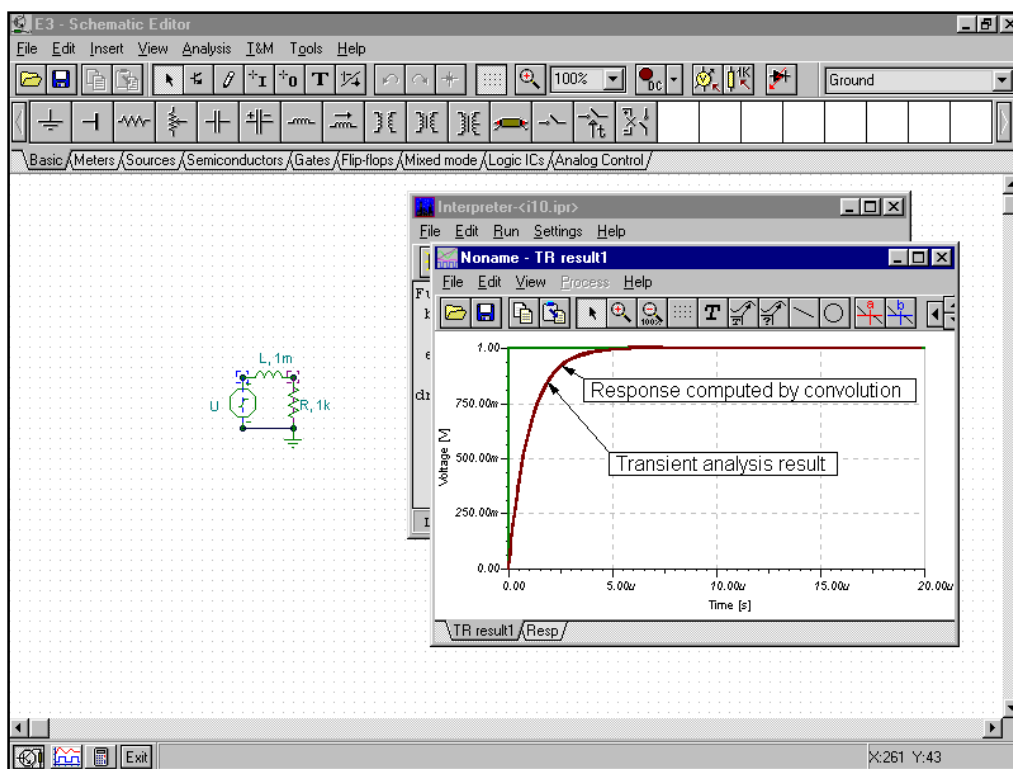
5.3 Verification

Let's use Transient Analysis on the circuit to check the result just found using the theoretical convolution function:

$$U_R(t) = 1 - e^{-\frac{R}{L}t}$$

We will check this result after running Transient Analysis by using the *copy-paste* functions in the *Diagram Window*. Moving to the Transient Analysis window, *TR_result1*, by clicking on the tab, we select the curve by clicking on it. We put it in the *Clipboard* by pressing the *Ctrl+C* key. Changing now to the *Resp* tab of the Diagram Window, we copy the selected curve into this diagram by pressing the *Ctrl+V* keys.

We can see that the results are the same—one overlays the other exactly.



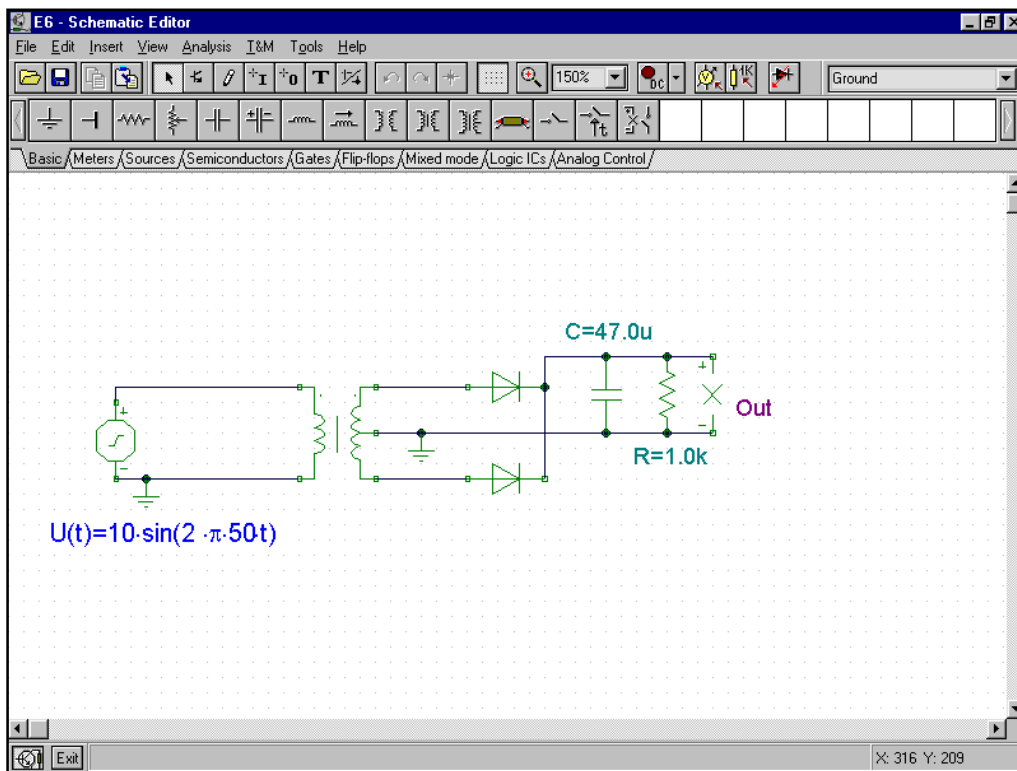
6 Processing TINA's results

Another interesting application of the Interpreter is post-processing the results of TINA analyses with arbitrary functions. When you carry out DC, AC, or transient analysis on a circuit in the Schematic Window, the analysis results also appear in the Interpreter symbol table, so you can reference them for additional processing.

6.1 Calculating average power

Example

Let's consider the following circuit. Determine the average power appearing in the resistor R, sensed by the voltage labeled *Out*, as a function of time.



Solution

The instantaneous power in the resistor is,

$$p(t) = u(t) \cdot i(t) = \frac{u^2(t)}{R}$$

The average power,

$$P(t) = \frac{1}{R \cdot t} \int_0^t p(s) ds$$

And,

$$P(t) = \frac{1}{R \cdot t} \int_0^t u^2(s) ds$$

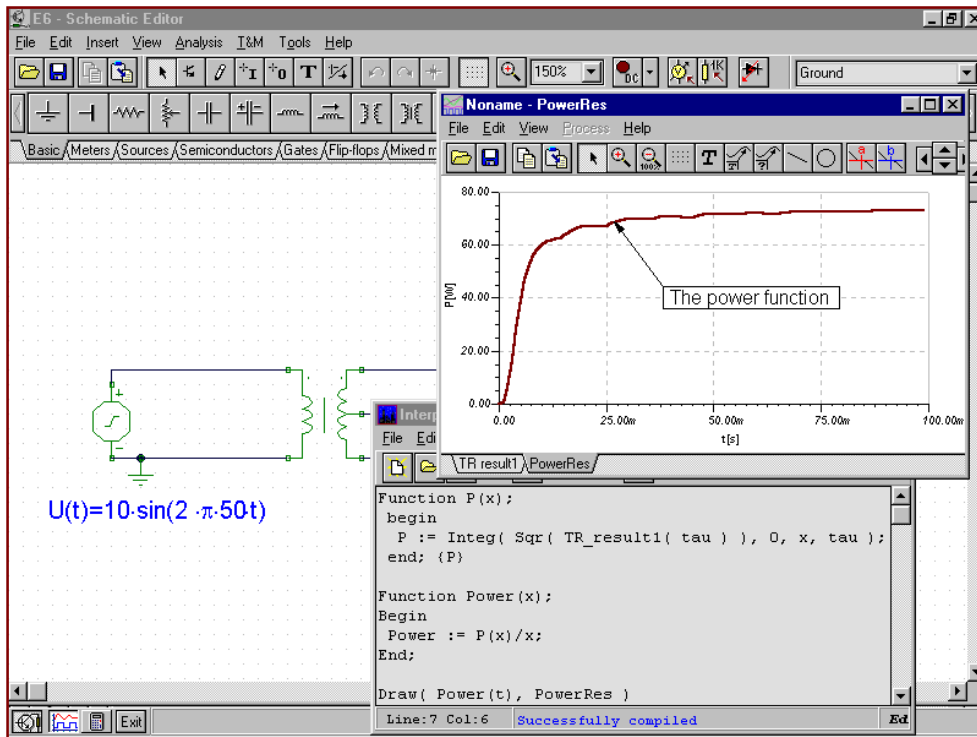
We will compute and draw the power function with the Interpreter. The $u(t)$ function in the integral (the voltage across the resistor) will be picked up from a previous TINA result produced by transient analysis if we reference it correctly, as TR_result1.

The solution as found via the Interpreter,

```
Function P(x);
begin
  P := Integ( Sqr( TR_result1( tau ) ), 0, x, tau );
end; {P}
```

```
Function Power(x);
begin
  Power := P(x) / (R*x);
end;
```

```
Draw( Power(t), PowerRes )
```



Verification

For verification we will use the fact that in this case the instantaneous voltage across the resistor settles down to an average amount with a ripple component, and the instantaneous and average values come out very close.

Using the TINA transient analysis output, let's check the result computed by the Interpreter.

In the TINA transient analysis time diagram, we see that after a while, the voltage across the resistor doesn't change much. This will let us use an average value which we will read off the curve in the Diagram Editor. Press the '*Cursor a*' button, and the cursor will change into a \oplus . Click this new cursor on the output curve, position it to around $80m$, and read the voltage:

$$U_{TRAN}(80m) = 8.4$$

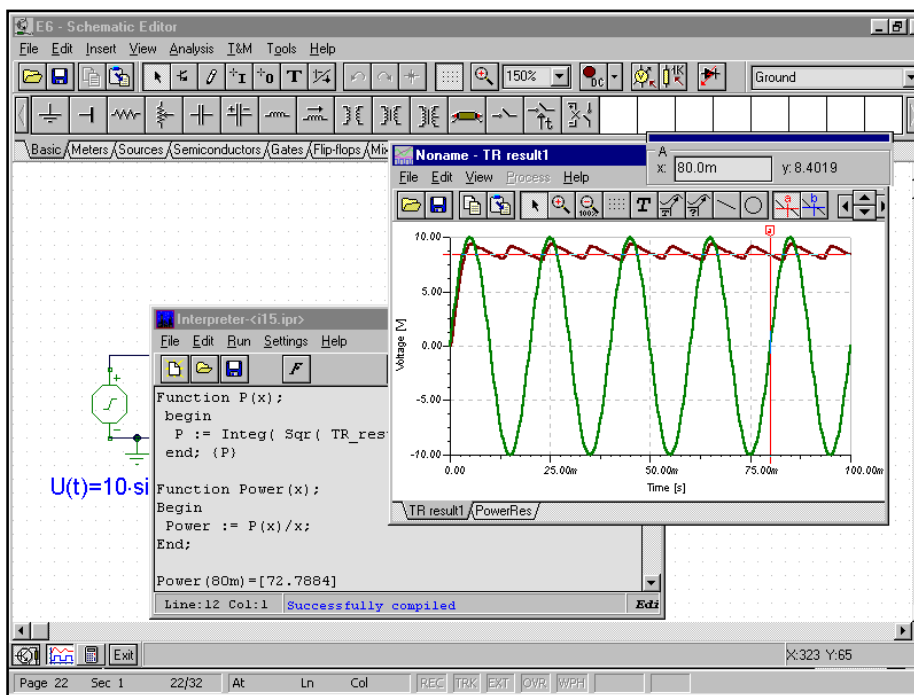
The average power at this time is:

$$p(80m) = \frac{u^2(80m)}{R} = 0.0705W = 70.05mW$$

Using the Interpreter we see that

$$\text{Power}(80m) = [72.7884]; \text{ that is, } 72.79mW$$

We find just a small difference of $2.74mW$.



Signal definition

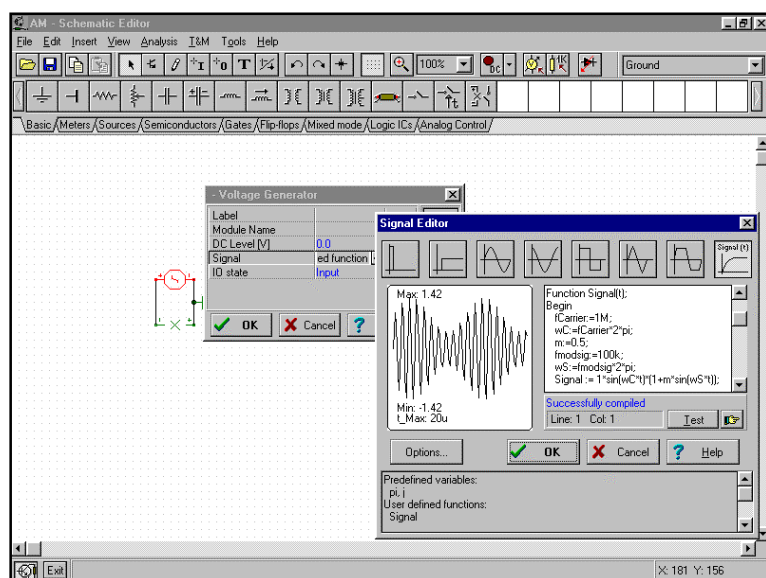
Another application of the Interpreter is the arbitrary definition of signals to be used as excitation for a circuit. These signals can be assigned to analog generators. User defined waveforms are extremely general and call up an equation editor window where the waveform may be defined as a mathematical expression. To make signal definition even easier, a template which you can modify to create your own signal is provided in the editor window. Note that while TINA's Interpreter is utilized for signal definition, you access signal definition via the Schematic Editor window, not the *Tools | Interpreter* menu.

By double-clicking on a generator, we can establish the *Signal* properties. Choose type *User defined* and the *Signal Editor* opens. In this editor, as in the Interpreter editor, we can edit, save and compile source code.

A separate window, visible just below the editor, contains a list of the symbols for pre-defined variables, circuit variables, and other user defined functions. We can use these to create a new user defined waveform. Here's how we would define an amplitude modulated signal:

```
Function Signal(t);
Begin
    fCarrier:=1M;
    wC:=fCarrier*2*pi;
    m:=0.5;
    fmodsig:=100k;
    wS:=fmodsig*2*pi;
    Signal := 1*sin(wC*t)*(1+m*sin(wS*t));
End;
```

If we continue and define more functions, the function named *Signal* will appear at the top of the symbol list. In the body of the functions, we can define variables and use arbitrary assignments of the *built-in* functions.



7 Using the Interpreter in the Training and Examination mode of TINA

For the educational user, *TINA* has two special modes. In examination mode, the student has to solve a series of problems (*a problem set*) either by traditional pencil-and-paper methods or by using the *Interpreter and analysis functions*. When the student finds the answer, the program sends it immediately to the teacher's machine, where it is promptly displayed by the TSuper supervision utility. In training mode, operation is similar, except that TINA gives the student feedback about the correctness of his answer, and the student may turn to the Advisor to get help prepared by the teacher.

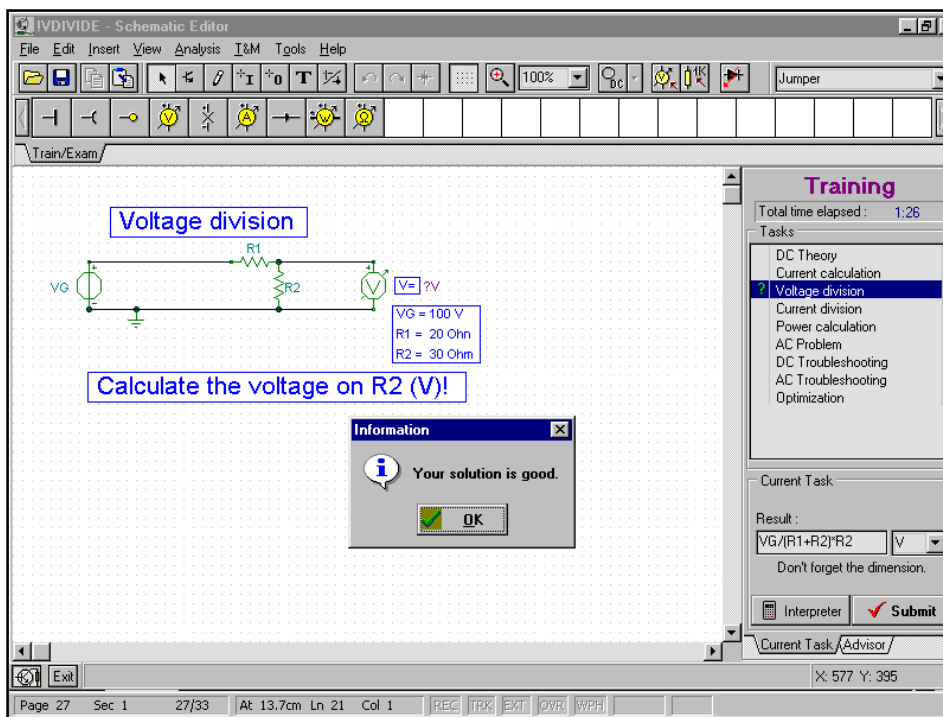
7.1 Solving a DC problem

The following example is a DC problem. The problem requires the student to calculate the voltage on the resistor R2. Clicking on the *Voltage division* entry in the examination panel, the circuit appears.

There are two ways to solve problems in TINA's training & examination mode. The first is by using the *Result field* in the Training & Examination panel, while the second is by using the *Interpreter* directly. Using the Interpreter directly, we can use the *Result* and the *Submit* functions to pass the result value to the training and examination module of TINA.

7.1.1 Using the Interpreter in the Result field

In the Result field, you may type in not only a number, but also a formula based on the symbols in the circuit. In this case, the Interpreter is automatically (internally) invoked and evaluates the formula.



7.1.2 Submitting a solution from the Interpreter window

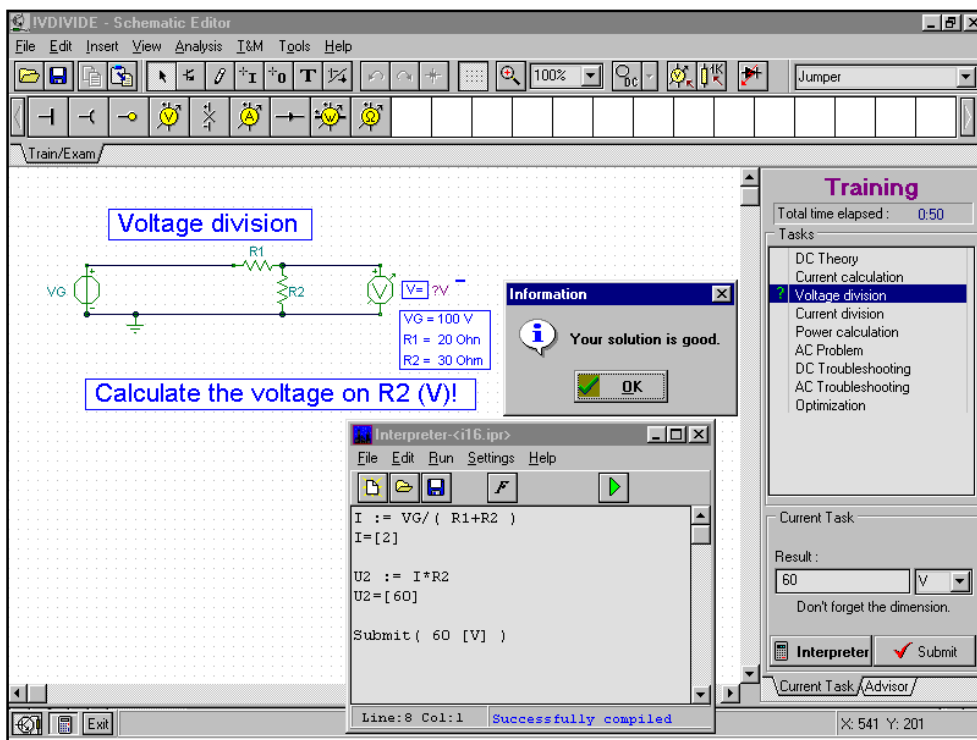
After clicking the Interpreter button, the Interpreter window opens. Remember that our example problem is to calculate the voltage on the resistor R2. We define two variables (I and U2) to hold the current and the voltage on R2 using Ohm's law. Using the *Submit* function, we pass the result to the examination module.

The solution with the Interpreter,

```
I := VG / ( R1+R2 )
I = [2]
```

```
U2 := I*R2
U2 = [60]
```

```
Submit ( 60 [V] )
```



7.2 Solving an AC problem

The following is an example of an AC problem. Click on the AC problem entry in the Examination and Training panel, and the circuit appears. Click the Interpreter button and the Interpreter window opens.

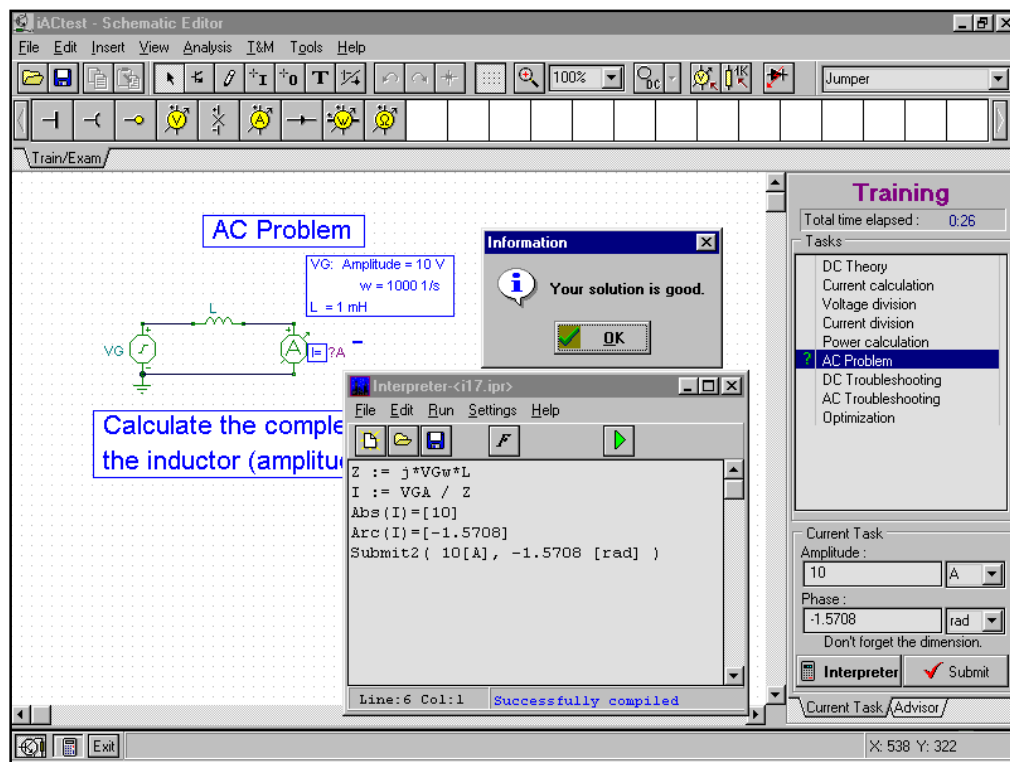
AC problems are normally too complex to allow a one-line solution in the result field, so we use the Interpreter Window. But if the expression is small enough, we could use the result field also.

The problem is calculating the complex current through the inductor. First we define two complex variables (Z and I) to hold the complex impedance and the complex current. Using the *Submit2* function we pass the result to the examination module.

The solution with the Interpreter

```
Z := j*VGw*L
I := VGA / Z
Abs(I) = [10]
Arc(I) = [-1.5708]
Submit2( 10[A], -1.5708 [rad] )
```

VGA stands for the voltage generator amplitude, while VGw stands for its radian frequency. It is good practice to review the results carefully before adding the Submit command to your solution. Once you press the Run button in a program containing the submit command, you are committed: TINA will check your solution, tell you if it is right or wrong, and submit it.



Appendix

There are a large number of *built-in* functions that you may use in the Interpreter. The formal parameters For some of these functions (identified with a C), the formal parameters accept complex numbers.

General purpose functions

Sin(x)

Calculates the sine of x

Cos(x)

Calculates the cosine of x

Tan(x)

Calculates the tangent of x

Atan(x)

Calculates the arc tangent of x

Exp(x)

Calculates the exponential function e^x

C

Ln(x)

Calculates the natural logarithm of x, $x > 0$

C

Sqr(x)

Calculates the square of x

C

Sqrt(x)

Calculates the positive square root of x

C

Abs(x)

Calculates the absolute value of x

C

Sgn(x)

Sign function:
$$\begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{if } x < 0 \end{cases}$$

Re(x)

Calculates the real part of x

C

Im(x)

Calculates the imaginary part of x

C

Arc(x)

Calculates the phase of x in radians

C

Sum(f(x))

Integral of f(x) in the [0..x] interval

Example:

`Sum(sin(pi)) = [2]`

Integ(Expression(τ),Limit1,Limit2,τ)

Integral of Expression(τ) in the [Limit1..Limit2] interval. The variable of integration is τ.

Example:

a)

```
Function Con(t);
Begin
  Con := t;
End;
Integ(Con(t), 0, 1, t) = [0.5]
```

b)

The meaning of `Integ(Con(t), 0, x, t)` is $\int_0^x Con(t)dt$.

D(f(x))

Derivative of f(x) at x

Example:

`D(sin(0)) = [1]`

Periodic(x,y)

Where y is the period value. The function transforms x into the [0..y] interval according to y.

Example:

`Periodic(5.3, 2) = [1.3]`

Not(x)

Calculates the bitwise negation of x

RadToDeg(x)

Converts x, in radians to degrees

DegToRad(x)

Converts x, in degrees to radians

Special functions for electrical engineering applications

Calculation of parallel impedances

Replus(x,y) = $\frac{x * y}{x + y}$ C

Unit step

$$\mathbf{E}(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ 1, & \text{if } x > 0 \end{cases}$$

Calculation of a value in decibels

$$\mathbf{Db}(x) = 20 * \lg(x), \quad x > 0$$

C

8 Index

- Abs function, 18**
- , 4**
- , 4, 17. See**
- Appendix, 12, 32**
- Bode diagrams, 17**
- Calculating the amplitude and the phase, 18**
- Calculations, 3**
- circuit parameters, 4**
- clipboard, 4**
- convolution, 22, 23**
- Diagram drawing, 15**
- Drawing impedances, 20**
- Drawing Preferences dialog, 6, 17, 20**
- Drawing time diagrams, 15**
- examination mode, 2, 7, 29**
- frequency domain, 17, 20**
- function declaration, 9**
- function definitions, 8**
- impedance function, 21**
- Integ function, 12**
- integrals, 2, 12**
- LOG-LIN diagram, 20**
- nodal equations, 14**
- PrintScreen, 4**
- Problem Solving, 7**
- processing, 25**
- Processing TINA's results, 25**
- Replus function, 21**
- Result field, 29**
- Signal processing, 5**
- Solving a DC problem, 29**
- Solving AC problems, 10**
- Solving an AC problem, 31**
- Solving DC problems, 8**
- Solving linear systems, 13**
- solving nodal systems, 13**
- Submit command, 31**
- symbol table, 5**
- systems of equations, 2**
- theoretical result, 15**
- transient analysis, 3, 5, 6, 15, 25, 26**
- user defined waveform, 7, 28**