# **ADVANTEST**

### **ADVANTEST CORPORATION**

TAS7x00Series Terahertz
Spectroscopic Imaging System
Math Library Reference Manual
(Automatic Control
Measuring Option)

**MANUAL NUMBER 8703117-02** 

Measuring Option)

### **Legal Notices**

All rights reserved. All text and figures included in this publication are the exclusive property of Advantest Corporation. Reproduction of this publication in any manner without the written permission of Advantest Corporation is prohibited. Information in this document is subject to change without notice.

#### Trademarks and Registered Trademarks

- ADVANTEST is a trademark of Advantest Corporation.
- Intel, Pentium, and Xeon are trademarks or registered trademarks of Intel Corporation in the United States and other countries.
- Microsoft, Windows, Windows 95, Windows 98, Windows 2000, Windows NT, Windows XP, Windows 7, Windows Server 2003, Excel, SQL Server, Visual C++, Visual C#, Visual Basic, and Visual Studio are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- LabVIEW is a trademark of National Instruments Corporation (United States).
- All other marks referenced herein are trademarks or registered trademarks of their respective owners.

Measuring Option)

# Revision History

Rev.	Date	Notes
01	Jan 10/14	
02	Mar 31/14	

# **Table of Contents**

Pre	efac	<b>Pre</b>	face-1
Р	urpo	se of This Manual Pre	face-1
Р	rered	quisites and Target UsersPre	face-1
S	amp	e Programs Used in This ManualPre	face-1
		ed Manuals Pre	
Ту	pog	raphic Convention Conver	ntion-1
С	onve	entions for Command Line and Program Examples Conver	ntion-1
Ir	ndica	tors for Operating the GUI Conver	ntion-1
Ir	ndica	tors for Keyboard OperationConver	ntion-2
R	efere	ence Source Notations Conver	ntion-2
U	nit Ir	dicators Conver	ition-2
1.	Ov	erview	1-1
1.	1	Functions of This Library	1-1
1.	2	Requirements for Using This Library	1-2
1.	3	Relationship Tables of TAS7500 System Software and Library Functions	1-3
			. 0
2.	Ins	stallation and Uninstall	2-1
2.	1	Installation	2-1
2.	2	Uninstall	2-4
3.	Ва	sic Usage of Arithmetic Library	3-1
3.	1	Process Flow	3-1
3.	2	Measure Function	3-2
3.	3	Analysis Function	3-3

. Calc	ulationResult Class
	1ember Functions
4. 1. 1	GetXValue
4. 1. 2	GetYValue
4. 1. 3	GetYValues
. AXIS	Conversion Class
5. 1 N	lember Functions
5. 1. 1	GetVersion
5. 1. 2	TASWindow
5. 1. 3	Apodization
5. 1. 4	FFT
5. 1. 5	PowerSpectrum
5. 1. 6	Transmittance
5. 1. 7	Absorbance
5. 1. 8	AbsorptionCoefficient
5. 1. 9	Phase
5. 1.10	PhaseShift
5. 1.11	CorrectPhaseShift
5. 1.12	ComplexRefractiveIndex
5. 1.13	Permittivity
. Caic	ulation Class
6. 1 N	lember Functions
6. 1. 1	GetVersion
6. 1. 2	Cut
6. 1. 3	SavitzkyGolaySmoothing
. Anal	ysis Class
7. 1 N	lember Functions
7. 1. 1	
	DetectPeak
append	ix 1.Error Messages

Appendix 2.Values of Constants	A2-1
List of Tables	T-1

### Preface

### **Purpose of This Manual**

This manual describes how to install and use the arithmetic library.

### **Prerequisites and Target Users**

This manual is written for those who have the following knowledge:

- The outline of the TAS7500 Series Terahertz Spectroscopy & Imaging System configuration
- Programming languages such as Excel VBA and LabVIEW

Target users

 System developers who analyze measurement results obtained with the TAS7500 Series Terahertz Spectroscopy & Imaging System.

### **Sample Programs Used in This Manual**

These sample programs are provided by Advantest to users free of charge as reference for creating software used on Advantest products. Users are allowed to copy or modify these sample programs to create new software. However, Advantest is not responsible for functionality issues with the software created by the user, problems with the created software, and damages due to such functionality issues or problems. By using these sample programs, users shall be deemed to have agreed to the above conditions.

### **Related Manuals**

TAS7x00Series Terahertz Spectroscopic Imaging System Remote API Reference Manual (Automatic Control Measuring Option)

This manual describes how to install the automatic control measurement option and use Remote APIs.

# **Typographic Convention**

### **Conventions for Command Line and Program Examples**

Command line and program examples are printed in a different font than the main body text so they can easily be distinguished. Similarly, different styles are used for user input and screen output.

Table C-1 Conventions for Command Line and Program Examples

Typeface or Symbol	Description	Example
1	Shows the [Return] having been pressed on a command line.	cd₊
abcd123	Command, directory, and file names, as well as screen output and program examples, are shown.	<pre>Set axisconv = CreateObject( "ATTAS.Math.Basic64.AxisConversion")</pre>
abcd123	When this typeface appears in a command line expression, it indicates the user must type in exactly the same.	dir.
[]	When this symbol appears in a syntax expression, the specification enclosed in [] can be omitted.	<pre>DetectPeak(source, peakType, smoothingPoints, positiveThreshold, negativeThreshold, sortType [, peaks])</pre>

### **Indicators for Operating the GUI**

The name of the option selection area of Graphical User Interface (GUI) screen is enclosed in the [] symbol. Also, operations for selecting commands within menus are indicated by an arrow symbol  $(\rightarrow)$ .

Table C-2 Indicators for Operating the GUI

Typeface or Symbol	Description	Example
[]	Shows the name of the option selection area on the GUI screen.	[File] menu [OK] button
[]→[]	Shows commands in a menu.	[File]→[Exit]

### **Indicators for Keyboard Operation**

Keyboard keys are enclosed in the [] symbol. When multiple keys are depressed simultaneously, the plus sign (+) is used as an indicator; multiple keys pressed in sequence are indicated by a comma.

Table C-3 Indicators for Keyboard Operation

Typeface or Symbol	Description	Example
[]	Shows keyboard keys.	[Return] key
[Ctrl]	Shows the [Control] key.	[Ctrl] key
+	Shows a second key pressed while a first one is held down when keys are connected by the plus (+) sign.	[Ctrl]+[d]
,	Shows each key sequentially when keys are delimited by commas.	[Ctrl]+[x], [y]

### **Reference Source Notations**

Reference sources are indicated by a change in typeface. Also, depending on the type of reference source, quotation marks or \* may be used.

Table C-4 Reference Source Notations

Typeface or symbol	Explanation	Example
"Reference Source Notations" on page Co nvention–2 Refer to Table C-4.	Indicates a chapter title, section title, or figure/table number. In the case of an online manual, clicking the string describing the reference source will cause the manual to jump to that page.	→ Refer to "Reference Source Notations" on page Convention-2 for information on notation methods for reference sources.  Table C-4 indicates the notation method for the reference source.
TAS7x00Series Tera- hertz Spectroscopic Imaging System Remote API Reference Manual (Automatic Control Measuring Option)	Indicates the title of the manual to which to refer. Clicking a manual title in an on-line manual will cause the document to jump to the "Related Manuals" on page Preface-1.	→ Refer to TAS7x00Series Terahertz Spectroscopic Imaging System Remote API Reference Manual (Automatic Control Measuring Option) for information on arithmetic libraries.

### **Unit Indicators**

Outside of the program examples and grammar discussed in this manual, SI unit symbols are used to indicate units.

# 1. Overview

This arithmetic library is for performing axis transformation, calculation, and analyses of measurement results obtained from the TAS7500 system.

### 1. 1 Functions of This Library

This library provides the calculations used by the TAS7500 system.

The major functions are the following:

- Function that performs axis transformation on waveform data
- Function that calculates waveform data, such as smoothing
- Function that analyzes waveform data, such as peak detection



All interfaces and sample programs using the functions described in this manual are written using Excel VBA.

# 1. 2 Requirements for Using This Library

The requirements for using this library are as follows:

• OS

Microsoft Windows 7 Professional [32 bit] Microsoft Windows 7 Professional [64 bit]

• Language

Excel VBA

LabVIEW version 8.2 or later

Visual Basic .NET

Visual C# .NET

• Required software

Microsoft .NET Framework 4.0 or later

# 1. 3 Relationship Tables of TAS7500 System Software and Library Functions

The functions provided by this library correspond to the menus for executing calculations of TAS7500 System Software. The following table shows those correspondences:

#### [Legend]

-: No corresponding function. (Conversion from Linear to Log is performed in the user program.)

N/A: No corresponding function. However, these functions are expected to correspond in the future.

#### Overview

### [Vertical Axis] menu

Vertical Axis	Measurement mode	Function name
Transmittance(Log) [dB]	Transmission	-
Transmittance(Linear) [%]	Transmission	Transmittance
Reflectance(Log) [dB]	Reflection	-
Reflectance(Linear) [%]	Reflection	N/A
ATR(Log) [dB]	ATR	-
ATR(Linear) [%]	ATR	N/A
Log(1/R)	Common	N/A
Absorbance	Transmission	Absorbance
Absorption Coefficient [cm-1]	Common	AbsorptionCoefficient
Extinction Coefficient	Transmission	ComplexRefractiveIndex
Extinction Coefficient	Reflection/ATR	N/A
Refractive Index	Transmission	ComplexRefractiveIndex
Refractive Index	Reflection/ATR	N/A
Permittivity	Transmission	Permittivity
Permittivity	Reflection/ATR	N/A
Dielectric Loss	Transmission	Permittivity
Dielectric Loss	Reflection/ATR	N/A
Phase Shift [rad]	Common	PhaseShift
Power(Log) [dB]	Common	-
Power(Linear)	Common	PowerSpectrum
Phase [rad]	Common	Phase
Time Domain [V]	Common	-
ATR Correction	ATR	N/A

### [Calculation] menu

Calculation	Function name
Scaling and Offset	N/A
Baseline Correction	N/A
Deconvolution	N/A
Derivative	N/A
FFT Filter	N/A
Interpolation	N/A
Normalize	N/A
Peak removal	N/A
Smoothing	SavitzkyGolaySmoothing

### [Analysis] menu

Analysis	Function name
Peak Detection	DetectPeak
Peak Height	N/A
Peak Area	N/A
Peak Half Bandwidth	N/A
Four Arithmetic Operations with Two Spectra	N/A

### [Polarization] menu

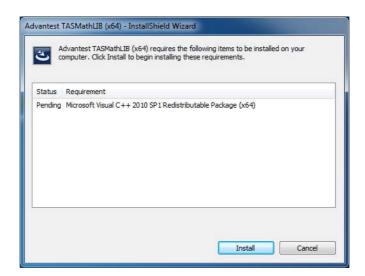
Polarization	Function name
All menus	N/A

# 2. Installation and Uninstall

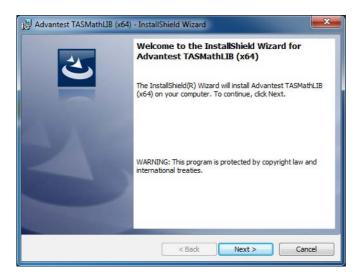
This chapter describes how to install and uninstall the arithmetic library.

### 2. 1 Installation

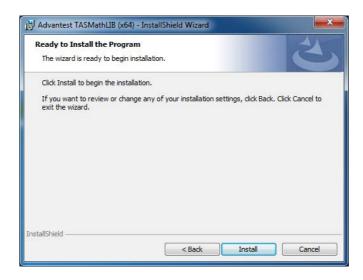
- 1) Insert the TAS7x00 System Software installation media (1/3) into the DVD-ROM drive.
- 2) When setup.bat in the TASMathLIB folder is double-clicked the User Account Control dialog box is displayed. Click the **[Allow]** button.
- 3) If Microsoft Visual C++ 2010 Redistributable Package has not been installed, the following window is displayed. Click the **[Install]** button. If it has already been installed, this window is not displayed.



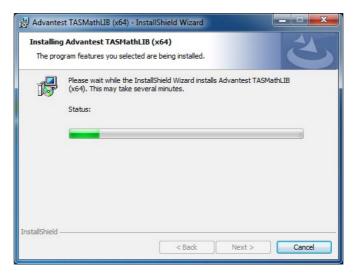
4) The following InstallShield Wizard window is then displayed:



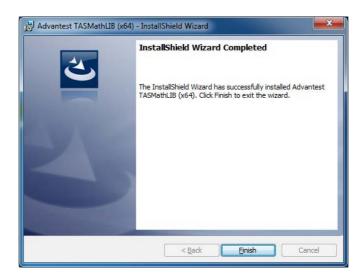
5) Click the **[Install]** button to start installing the software.



6) A window showing the installation progress is then displayed.

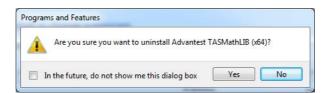


7) When the installation is complete, click the **[Finish]** button.



### 2. 2 Uninstall

- 1) Select [Control Panel] from the Windows [Start] menu, and then open [Programs and Features].
- 2) Select Advantest TASMathLIB from the list and click [Uninstall].
- 3) When the following confirmation message box is displayed, click the **[OK]** button:



4) The User Account Control dialog box is then displayed. Click the **[Continue]** button. This completes uninstall of this option.

# 3. Basic Usage of Arithmetic Library

This chapter describes the basic usage of the arithmetic library.

### 3. 1 Process Flow

This section describes the flow from obtaining time waveform data using the TAS7x00 system to detecting peaks after converting the time waveform data into absorbance data, by using a sample Excel VBA program.

In the following sample program, time waveform data is obtained by using the Measure function, and the time waveform data is converted into absorbance data by using the Analysis function to detect peaks. Details of these two functions are described in the subsequent sections.

Sample program (Main function)

```
' Reference data
Dim Background
' Sample data
Dim Sample

Sub Main()
    ' Performs sample measurement to obtain time waveform data.
    Call Measure

    ' Calculates the absorbance and executes smoothing, and then obtains peaks.
    Call Analysis
End Sub
```

### 3. 2 Measure Function

To obtain time waveform data using the TAS7500 system, a Remote API is used.

→ For more information on Remote APIs, refer to TAS7x00Series Terahertz Spectroscopic Imaging System Remote API Reference Manual (Automatic Control Measuring Option).

In the Measure function of the sample program, reference time waveform data is stored in the Background variable and sample time waveform data is stored in the Sample variable.

Sample program (Measure function)

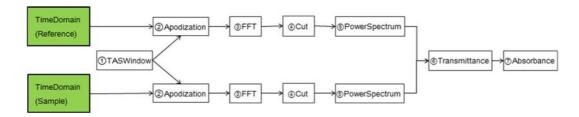
```
Sub Measure()
   Set api = CreateObject("ATTAS.Remote.API")
    ' Connects to the TAS7500 system.
   Call api.Connect("127.0.0.1")
    ' Resets the TAS7500 system.
   Call api.Reset
    ' Performs background measurement by specifying a measurement units, frequency
     resolution, and cumulated number.
   Call api.MeasureBackground(1, 7.6, 1024)
    ' Obtains background data as time waveform data.
   Set Background = api.GetBackground(0)
    ' Performs sample measurement by specifying a measurement unit, frequency
      resolution, and cumulated number.
   Call api.MeasureSample(1, 7.6, 1024)
    ' Obtains sample data as time waveform data.
    Set Sample = api.GetSample(0)
End Sub
```

In this sample program, time waveform data is obtained. However, the power spectrum can also be obtained with the GetBackground function or GetSample function of Remote APIs.

### 3. 3 Analysis Function

The flow for calculating absorbance data from time waveform data is as follows.

The green boxes represent time waveform data and white ones represent functions.



The absorbance is obtained by performing the following calculations on sample data and reference time waveform data:

- 1. Create window function data.
- 2. Apply the window function data to time waveform data.
- 3. Execute fast Fourier transform to obtain frequency data.
- 4. Because the effective frequency band varies depending on the optical system being used, extract the desired frequency range from the frequency data.
- 5. Create a power spectrum.
- 6. Calculate transmittance from the sample power spectrum and reference power spectrum. (When the power spectrum has been obtained by using a Remote API, steps 1. to 5. are omitted and the flow starts from this step.)
- 7. Calculate absorbance from the transmittance.

In the Analysis function of the sample program, peak detection is performed by smoothing after calculating absorbance data in accordance with the preceding flow. The results are output to Sheet 1.



The arithmetic library has a Microsoft COM (Component Object Model) interface. When programming, reference the COM object corresponding to each language.

#### 3. Basic Usage of Arithmetic Library

#### Sample program (Analysis function)

```
Sub Analysis()
    Set axisconv = CreateObject("ATTAS.Math.Basic32.AxisConversion")
    Set analy = CreateObject("ATTAS.Math.Basic32.Analysis")
    Set calc = CreateObject("ATTAS.Math.Basic32.Calculation")
    ' Calculates the window function.
    win = axisconv.TASWindow(65536)
    ' Applies the window function.
    Set timeBak = axisconv.Apodization(Background.GetXValue,
        Background.GetYValue, win)
    Set timeSam = axisconv.Apodization(Sample.GetXValue, Sample.GetYValue, win)
    ' Executes fast Fourier transform.
    Set freqBak = axisconv.FFT(timeBak.GetXValue, timeBak.GetYValue(0))
    Set freqSam = axisconv.FFT(timeSam.GetXValue, timeSam.GetYValue(0))
    ' Cuts out the waveform data.
    Set bak = calc.Cut(freqBak.GetXValue, freqBak.GetYValues, 0, 5)
    Set sam = calc.Cut(freqSam.GetXValue, freqSam.GetYValues, 0, 5)
    ' Calculates the power spectrum.
    Set powerBak = axisconv.PowerSpectrum(bak.GetXValue, bak.GetYValues)
    Set powerSam = axisconv.PowerSpectrum(sam.GetXValue, sam.GetYValues)
    ' Calculates the transmittance.
    Set trans = axisconv.Transmittance(powerSam.GetXValue,
        powerSam.GetYValue(0),powerBak.GetYValue(0))
    ' Calculates the absorbance.
    Set abso = axisconv.Absorbance(trans.GetXValue, trans.GetYValue(0))
    ' Executes smoothing.
    Set savgol = calc.SavitzkyGolaySmoothing(abso.GetXValue,
        abso.GetYValue(0), 101, 3)
    ' Calculates peaks.
    y = savgol.GetYValue(0)
    peaks = analy.DetectPeak(y, 0, 3, 0.01, -0.01, 0)
    ' Enter the obtained data in the cells of Sheet 1.
    For Each peak In peaks
       Sheet1.Cells(idx, 1) = y(peak)
        idx = idx + 1
    Next
End Sub
```



There are two arithmetic libraries, 32-bit and 64-bit arithmetic libraries. Select one according to the software that executes the library function. In the preceding sample program, the 32-bit arithmetic library is specified for the argument of the CreateObject function.

# 4. CalculationResult Class

### 4. 1 Member Functions

This section describes the member functions of the CalculationResult class.

Namespace: ATTAS.Math.Basic32.CalculationResult

ATTAS.Math.Basic64.CalculationResult

### 4. 1. 1 GetXValue

This function returns the X-axis data array of calculation results.

### ♦ Syntax

<Excel VBA>

GetXValue()

#### ◆ Return value

<Excel VBA>

Return value type	Description	
Double()	X-axis data array of calculation results.	

### 4. 1. 2 GetYValue

This function returns the Y-axis data array of calculation results.

### ◆ Syntax

<Excel VBA>

GetYValue(index)

#### 4. CalculationResult Class

### ◆ Argument

<Excel VBA>

Argument name	I/O	Data type	Description
index	In	Integer	Multidimensional array element number.

#### ◆ Return value

<Excel VBA>

Return value type	Description
Double()	Y-axis data array of calculation results.

### ◆ Error conditions

1) The index is out of the range.

### 4. 1. 3 GetYValues

This function returns the Y-axis data of calculation results in a multidimensional array.

### ♦ Syntax

<Excel VBA>

GetYValues

#### ◆ Return value

<Excel VBA>

Return value type	Description
Double(,)	Y-axis multidimensional data array (Y axis) of calculation results.

# 5. AxisConversion Class

### 5. 1 Member Functions

This section describes the member functions of the AxisConversion class.

Namespace: ATTAS.Math.Basic32.AxisConversion

ATTAS.Math.Basic64.AxisConversion

### 5. 1. 1 GetVersion

This function returns the arithmetic library version.

### ♦ Syntax

<Excel VBA>

GetVersion()

#### ◆ Return value

<Excel VBA>

Return value type	Description	
String	Arithmetic library version (example: "R1.0").	

### 5. 1. 2 TASWindow

This function creates a window function data array.

### ♦ Syntax

<Excel VBA>

TASWindow(length)

#### 5. AxisConversion Class

### ◆ Argument

<Excel VBA>

Argument name	I/O	Data type	Description
length	In	Integer	Data array element length of the window function to be created. Specify a power of two.

#### ◆ Return value

<Excel VBA>

Return value type	Description	
Double()	Window function data array.	

#### ◆ Details

Generates a  $exp(-x^8)$  -type data array in which the peak is at 1/8 of the data length and the amplitude drops down to 0.000001 at the beginning and end of the data.

#### ◆ Error conditions

- 1) The length is less than 1024.
- 2) The length is not a power of two.

### 5. 1. 3 Apodization

This function applies the window function to time waveform.

### ◆ Syntax

<Excel VBA>

Apodization (timeDomainX, timeDomainY, window)

### ◆ Argument

<Excel VBA>

Argument name	I/O	Data type	Description
timeDomainX	In	Double()	Time waveform data array (X axis) [psec].
timeDomainY	In	Double()	Time waveform data array (Y axis) [V].
window	In	Double()	Window function data array.

#### ◆ Return value

<Excel VBA>

Return value type	Description		
CalculationResult	Calculation result data array (X axis and Y axis).		
	X[n]: Time waveform data array [psec]. $Y[0,n]$ : Time waveform data array after the window function is applied [V].		

#### ◆ Details

Integrates the time waveform data array (Y axis) and the window function data array for each element.

#### ◆ Error conditions

- 1) One of timeDomainX, timeDomainY, and window is NULL or the number of elements is 0.
- 2) The number of elements is different among timeDomainX, timeDomainY, and window.

### 5. 1. 4 FFT

This function executes fast Fourier transform.

### ◆ Syntax

<Excel VBA>

FFT(timeDomainX, timeDomainY)

#### 5. AxisConversion Class

### ◆ Argument

<Excel VBA>

Argument name	I/O	Data type	Description
timeDomainX	In	Double()	Time waveform data array (X-axis) [psec]. Specify a data array whose number of elements is a power of two.
timeDomainY	In	Double()	Time waveform data array (Y axis) [V]. Specify a data array whose number of elements is a power of two.

#### ◆ Return value

<Excel VBA>

Return value type	Description
CalculationResult	Calculation result data array (X axis and Y axis).  X[n]: Frequency data array [THz] Y[0, n]: FFT data array (real part) Y[1, n]: FFT data array (imaginary part)

#### ◆ Details

Executes fast Fourier transform on time waveform.

#### ◆ Error conditions

- 1) timeDomainX or timeDomainY is NULL or the number of elements is 0.
- 2) The number of elements is different between timeDomainX and timeDomainY.
- 3) The number of elements of timeDomainX or timeDomainY is not a power of two.

### 5. 1. 5 PowerSpectrum

This function calculates the power spectrum.

### ◆ Syntax

<Excel VBA>

PowerSpectrum(frequencyX, fftY)

### ◆ Argument

<Excel VBA>

Argument name	I/O	Data type	Description
frequencyX	In	Double()	Frequency data array (X axis) [THz].
fftY	In	Double(,)	FFT data array (Y axis)
			fftY[0,n]: Real part fftY[1,n]: Imaginary part

#### ◆ Return value

<Excel VBA>

Return value type	Description	
CalculationResult	Calculation result data array (X axis and Y axis).	
	X[n]: Frequency data array [THz] Y[0,n]: Power spectrum data array	

#### ◆ Details

The power spectrum is calculated using the following expressions:

$$P_{\mathit{Linear}}(\omega) = \left|\widetilde{F}(\omega)\right|^2$$

 $P_{\mathit{Linear}}(\omega)$  : Power spectrum

 $\widetilde{F}(\omega)$  : FFT data (complex number)

 $|\widetilde{F}(\omega)| = \sqrt{(\widetilde{F}(\omega).real)^2 + (\widetilde{F}(\omega).image)^2}$  : Absolute value of the complex number

#### ◆ Error conditions

- 1) frequencyX or fftY is NULL or the number of elements is 0.
- 2) The number of elements is different between frequencyX and fftY.

### 5. 1. 6 Transmittance

This function calculates the transmittance.

#### 5. AxisConversion Class

### ♦ Syntax

<Excel VBA>

Transmittance(sampleFrequencyX, samplePowerLinearY, referencePowerLinearY)

### Argument

<Excel VBA>

Argument name	I/O	Data type	Description
sampleFrequencyX	In	Double()	Sample frequency data array (X axis) [THz].
samplePowerLinearY	In	Double()	Sample power spectrum data array (Y axis).
referencePowerLinearY	In	Double()	Reference power spectrum data array (Y axis).

#### ◆ Return value

<Excel VBA>

Return value type	Description	
CalculationResult	Calculation result data array (X axis and Y axis).	
	X[n]: Sample frequency data array [THz] Y[0,n]: Transmittance data array [%]	

#### ◆ Details

The transmittance is calculated using the following expressions:

$$T_{Linear}(\omega) = \frac{P_{sam}(\omega)}{P_{ref}(\omega)} \times 100$$

 $T_{\it Linear}(\omega)$  : Transmittance [%]

 $P_{sam}(\omega)$  : Sample power spectrum

 $P_{\mathit{ref}}(\omega)$  : Reference power spectrum

#### ◆ Error conditions

- 1) One of sampleFrequencyX, samplePowerLinearY, and referencePowerLinearY is NULL or the number of elements is 0.
- 2) The number of elements is different among sampleFrequencyX, samplePowerLinearY, and referencePowerLinearY.

## 5. 1. 7 Absorbance

This function calculates the absorbance.

#### ◆ Syntax

<Excel VBA>

Absorbance(frequencyX, transmittanceLinearY)

### ◆ Argument

<Excel VBA>

Argument name	I/O	Data type	Description
frequencyX	In	Double()	Frequency data array (X axis) [THz].
transmittanceLinearY	In	Double()	Transmittance data array (Y axis) [%].

#### ◆ Return value

<Excel VBA>

Return value type	Description		
CalculationResult	Calculation result data array (X axis and Y axis).		
	X[n]: Frequency data array [THz] Y[0,n]: Absorbance data array		

#### Details

The absorbance is calculated using the following expressions:

$$A(\omega) = -Log_{10} \left[ \frac{T_{Linear}(\omega)}{100} \right]$$

 $A(\omega)$  : Absorbance

 $T_{\it Linear}(\omega)$  : Transmittance [%]

- 1) frequencyX or transmittanceLinearY is NULL or the number of elements is 0.
- 2) The number of elements is different between frequencyX and transmittanceLinearY.

# 5. 1. 8 AbsorptionCoefficient

This function calculates the absorption coefficient.

#### ◆ Syntax

<Excel VBA>

AbsorptionCoefficient(frequencyX, extinctionCoefficientY)

## Argument

<Excel VBA>

Argument name	I/O	Data type	Description
frequencyX	In	Double()	Frequency data array (X axis) [THz].
extinctionCoefficientY	In	Double()	Extinction coefficient data array (Y axis).

#### ◆ Return value

<Excel VBA>

Return value type	Description	
CalculationResult	Calculation result data array (X axis and Y axis).  X[n]: Frequency data array [THz]	
	Y[0,n]: Absorption coefficient data array.	

#### Details

The absorption coefficient is calculated using the following expressions:

$$\alpha(\omega) = \frac{2\omega\kappa(\omega)}{c} \times \frac{10^{12}}{100}$$

 $\alpha(\omega)$ : Absorption coefficient [cm-1]

 $\kappa(\omega)$ : Extinction coefficient

ω : Angular frequency [rad/s]

c : Speed of light [m/s]

#### ◆ Error conditions

1) frequencyX or extinctionCoefficientY is NULL or the number of elements is 0.

2) The number of elements is different between frequencyX and extinctionCoefficientY.

# 5. 1. 9 Phase

This function calculates the phase.

# ♦ Syntax

<Excel VBA>

Phase(frequencyX, fftY [, thresholdAngle [, startFrequency [, initialValue]]])

# ◆ Argument

<Excel VBA>

Argument name	I/O	Data type	Description
frequencyX	In	Double()	Frequency data array (X axis) [THz].
fftY	In	Double(,)	FFT data array (Y axis).  Y[0,n]: Real part  Y[1,n]: Imaginary part
thresholdAngle	In	Integer	Angle criterion for unwrap processing [deg] (0 to 360). Default value: 210
startFrequency	In	Double	Start frequency for unwrap processing [THz] (last value from 0 to frequencyX).  Default value: 0.2
initialValue	In	Integer	Initial value for unwrap processing (0 to 100).  Default value: 0

### ◆ Return value

Return value type	Description	
CalculationResult	Calculation result data array (X axis and Y axis).  X[n]: Frequency data array [THz] Y[0,n]: Phase data array [rad]	

#### ◆ Details

The phase is calculated using the following expressions:

$$\theta(\omega) = -Arg\big[\widetilde{F}(\omega)\big]$$

 $\theta(\omega)$ : Phase [rad]

 $\widetilde{F}(\omega)$ : FFT data (complex number)

Arg[]: Deflection angle  $tan^{-1}\left(\frac{y}{x}\right)$ 

#### ◆ Error conditions

- 1) frequencyX or fftY is NULL or the number of elements is 0.
- 2) The number of elements is different between frequencyX and fftY.
- 3) thresholdAngle is out of the range.
- 4) startFrequency is out of the range.
- 5) initialValue is out of the range.

### 5. 1.10 PhaseShift

This function calculates the phase difference.

#### ◆ Syntax

<Excel VBA>

PhaseShift(sampleFrequencyX, samplePhaseY, referencePhaseY)

## Argument

Argument name	I/O	Data type	Description
sampleFrequencyX	In	Double()	Sample frequency data array (X axis) [THz].
samplePhaseY	In	Double()	Sample phase data array (Y axis) [rad].
referencePhaseY	In	Double()	Reference phase data array (Y axis) [rad].

#### ◆ Return value

<Excel VBA>

Return value type	Description		
CalculationResult	Calculation result data array (X and Y axes).		
	X[n]: Frequency data array [THz] Y[0,n]: Phase difference data array [rad]		

#### ◆ Details

The phase difference is calculated using the following expressions:

$$\phi(\omega) = \theta_{sam}(\omega) - \theta_{ref}(\omega)$$

$$\phi(\omega) : \text{Phase difference [rad]}$$

$$\theta_{sam}(\omega) : \text{Sample phase [rad]}$$

$$\theta_{ref}(\omega) : \text{Reference phase [rad]}$$

## ◆ Error conditions

- 1) One of sampleFrequencyX, samplePhaseY, and referencePhaseY is NULL or the number of elements is 0.
- 2) The number of elements is different among sampleFrequencyX, samplePhaseY, and referencePhaseY.

### 5. 1.11 CorrectPhaseShift

This function executes phase difference compensation.

## ◆ Syntax

<Excel VBA>

CorrectPhaseShift(frequencyX, phaseShiftY, sampleTimeDomainX, sampleTimeDomainY,
referenceTimeDomainY [, predictionFrequency [, availableRange ]])

## ◆ Argument

<Excel VBA>

Argument name	I/O	Data type	Description
frequencyX	In	Double()	Frequency data array (X axis) [THz].
phaseShiftY	In	Double()	Phase difference data array (Y axis) [rad].
sampleTimeDomainX	In	Double()	Sample time waveform data array (X axis) [psec].
sampleTimeDomainY	In	Double()	Sample time waveform data array (Y axis) [V].
referenceTimeDomainY	In	Double()	Reference time waveform data array (Y axis) [V].
predictionFrequency	In	Double	Correction frequency [THz] (0 to 10). Default value: 0.2
availableRange	In	Double	Valid range width (0 or larger). Default value: 0.2

#### ◆ Return value

<Excel VBA>

Return value type	Description	
CalculationResult	Calculation result data array (X axis and Y axis).	
	X[n]: Frequency data array [THz] Y[0,n]: Corrected phase difference data array [rad]	

#### ◆ Details

Corrects phase difference data in a low-frequency area so that it becomes a value close to the expected value of the phase difference calculated from time domain data at a specified frequency.

- 1) One of frequencyX, phaseShiftY, sampleTimeDomainX, sampleTimeDomanY, and reference-TimeDomainY is NULL or the number of elements is 0.
- 2) The number of elements is different between frequency X and phase Shift Y.
- 3) The number of elements is different among sampleTimeDomainX, sampleTimeDomanY, and referenceTimeDomainY.
- 4) predictionFrequency is out of the range.
- 5) availableRange is out of the range.

# 5. 1.12 ComplexRefractiveIndex

This function calculates the refractive index and extinction coefficient.

## ◆ Syntax

<Excel VBA>

ComplexRefractiveIndex(frequencyX, transmittanceLinearY, phaseShiftY,
sampleThickness [, initialReal [, initialImaginary [, repeatCount [,
multipleReflection]]]])

# ◆ Argument

<Excel VBA>

Argument name	I/O	Data type	Description
frequencyX	In	Double()	Frequency data array (X axis) [THz].
transmittanceLinearY	In	Double()	Transmittance data array (Y axis) [%].
phaseShiftY	In	Double()	Phase difference data array (Y axis) [rad].
sampleThickness	In	Double	Sample thickness [mm]. (0.00001 to 10.0)
initialReal	In	Double	Complex refractive index initial value (refractive index). Default value: 1.0
initialImaginary	In	Double	Complex refractive index initial value (extinction coefficient).  Default value: 0.001
repeatCount	In	Integer	Number of calculation repetitions (0 or larger). Default value: 10
multipleReflection	In	Integer	Number of multiple reflections (0 to 50). Default value: 0

#### ◆ Return value

Return value type	Description		
CalculationResult	Calculation result data array (X axis and Y axis).		
	X[n]: Frequency data array [THz] Y[0, n]: Refractive index data array Y[1, n]: Extinction coefficient data array		

#### Details

The refractive index and extinction coefficient are calculated using the following expressions. In expressions ① and ②, the assignment operation is repeated as many times as the specified number of calculation repetitions from the initial value of  $n(\omega)$  and  $\kappa(\omega)$ .

$$\widetilde{n}(\omega) = n(\omega) - i\kappa(\omega)$$

$$\widetilde{t}_{as}(\omega) = \frac{2}{\widetilde{n}(\omega) + 1} \qquad \widetilde{t}_{as}(\omega) = \frac{2}{\widetilde{n}(\omega) + 1} \qquad \widetilde{r}_{sa}(\omega) = \widetilde{r}_{as}(\omega) = \frac{\widetilde{n}(\omega) - 1}{\widetilde{n}(\omega) + 1}$$

$$n(\omega) = \frac{c}{d\omega} \left[ \phi(\omega) + \frac{d\omega}{c} + Arg \left[ \widetilde{t}_{cs}(\omega) \widetilde{t}_{sa}(\omega) \times \sum_{l=0}^{m} \left( (\widetilde{r}_{sa}(\omega))^{2} \exp\left[ -i \frac{2\widetilde{n}(\omega)d\omega}{c} \right] \right)^{l} \right] \right] \dots 1$$

$$\kappa(\omega) = -\frac{c}{2d\omega} \times ln \left[ \frac{T_{Linear}(\omega)}{\left| \widetilde{t}_{as}(\omega) \widetilde{t}_{sa}(\omega) \sum_{l=0}^{m} \left( (\widetilde{r}_{sa}(\omega))^{2} exp \left[ -i \frac{2\widetilde{n}(\omega)d\omega}{c} \right] \right)^{l} \right|^{2}} \times \frac{1}{100} \right] \dots 2$$

 $\widetilde{n}(\omega)$ : Complex index of refraction,  $n(\omega)$ : complex refractive index,  $\kappa(\omega)$ : extinction coefficient

c: Speed of light [m/s], d: sample thickness [m],  $\phi(\omega)$ : phase difference

 $T_{\it Linear}(\omega)$ : Transmittance [%], m: number of multiple reflections,  $\it Arg[$ ]: deflection angle  $\it tan^{-1}(\frac{y}{x})$ 

- 1) One of frequencyX, transmittanceLinearY, and phaseShiftY is NULL or the number of elements is 0.
- 2) The number of elements is different among frequency X, transmittance Linear Y, and phase Shift Y.
- 3) sampleThickness is out of the range.
- 4) repeatCount is out of the range.
- 5) multipleReflection is out of the range.

# 5. 1.13 Permittivity

This function calculates the permittivity and dielectric loss.

# ♦ Syntax

<Excel VBA>

Permittivity(frequencyX, complexRefractiveIndexY)

# ◆ Argument

<Excel VBA>

Argument name	I/O	Data type	Description
frequencyX	In	Double()	Frequency data array (X axis) [THz].
complexRefractiveIndexY	In	Double(,)	Complex refractive index data array (Y axis). complexRefractiveIndexY [0,n]: Refractive index data array complexRefractiveIndexY [1,n]: Extinction coefficient data array

### ◆ Return value

Return value type	Description
CalculationResult	Calculation result data array (X axis and Y axis).  X[n]: Frequency data array [THz] Y[0, n]: Permittivity Y[1, n]: Dielectric loss

#### ◆ Details

The permittivity and dielectric loss are calculated using the following expressions:

$$\varepsilon_1(\omega) = n(\omega)^2 - \kappa(\omega)^2$$

$$\varepsilon_2(\omega) = 2n(\omega)\kappa(\omega)$$

 $\varepsilon_1(\omega)$ : Permittivity

 $\varepsilon_2(\omega)$ : Dielectric loss

 $n(\omega)$ : Refractive index

 $\kappa(\omega)$ : Extinction coefficient

- 1) frequencyX or complexRefractiveIndexY is NULL or the number of elements is 0.
- 2) The number of elements is different between frequencyX and complexRefractiveIndexY.

# 6. Calculation Class

# 6. 1 Member Functions

This section describes the member functions of the Calculation class.

Namespace: ATTAS.Math.Basic32.Calculation

ATTAS.Math.Basic64.Calculation

### 6. 1. 1 GetVersion

This function returns the arithmetic library version.

# ♦ Syntax

<Excel VBA>

GetVersion()

#### ◆ Return value

<Excel VBA>

Return value type	Description	
String	Arithmetic library version (example: "R1.0").	

### 6. 1. 2 Cut

This function extracts the specified range from waveform data.

### ◆ Syntax

<Excel VBA>

Cut(sourceX, sourceY, startX, endX)

#### 6. Calculation Class

### ◆ Argument

<Excel VBA>

Argument	I/O	Data typ	Description
sourceX	In	Double()	Waveform data array (X axis).
sourceY	In	Double(,)	Waveform data array (Y axis).  Y[0,n]: Waveform data array 1 Y[1,n]: Waveform data array 2 Y[m,n]: Waveform data array m
startX	In	Double	Extraction start position (X axis).
endX	In	Double	Extraction end position (X axis).

#### ♦ Return value

<Excel VBA>

Return value type	Description	
CalculationResult	Calculation result data array (X and Y axes).	
	X[n]: Waveform data array (X axis) Y[0,n]: Extracted waveform data array 1 Y[1,n]: Extracted waveform data array 2 Y[m,n]: Extracted waveform data array m	

#### ◆ Details

Extracts multiple waveform data items at once. To do so, specify multiple waveform data items for sourceY.

If the range specified for startX and endX is out of the sourceX range, an empty array is returned.

#### ◆ Error conditions

- 1) sourceX or sourceY is NULL or the number of elements is 0.
- 2) The number of elements is different between sourceX and sourceY.
- 3) startX > endX.

# 6. 1. 3 SavitzkyGolaySmoothing

This function performs smoothing using the Savitzky-Golay method.

6. 1 Member Functions

## ◆ Syntax

<Excel VBA>

SavitzkyGolaySmoothing(sourceY, sourceY, smoothingPoints, polynomialOrder)

### Argument

<Excel VBA>

Argument name	I/O	Data type	Description
sourceX	In	Double()	Waveform data array (X axis).
sourceY	In	Double()	Waveform data array (Y axis).
smoothingPoints	In	Integer	Neighborhood points used for smoothing (odd numbers from 3 to the number of sourceX elements).
polynomialOrder	In	Integer	Polynomial order (1 to 5).

#### ◆ Return value

<Excel VBA>

Return value type	Description	
CalculationResult	Calculation result data array (X and Y axes).	
	X[n]: Waveform data array (X axis). Y[0,n]: Smoothed waveform data array (Y axis)	

#### Details

The Savitzky-Golay method is a type of moving average method, which approximates waveform data to the polynomial curve in the neighborhood of each sample point. This method is expressed using the following expressions:

$$g_i = \sum_{n=-n_L}^{n_R} C_n f_{i+n}$$
  $(i = 0,1,...,N-1)$ 

 $g_i$  : Smoothing waveform,  $f_{i+n}$  : Waveform data,  $c_n$  : Savitzky-Golay coefficient,

 $n_{\!\scriptscriptstyle L}$  : Left neighborhood point,  $n_{\!\scriptscriptstyle R}$  : Right neighborhood point, N : Number of elements of the waveform data

#### 6 Calculation Class

Here, the coefficient  $c_n$  is calculated as follows:

Design matrix 
$$A_{ij} = i^j$$
  $(i = -n_L, ..., n_R, )$   $(j = 0, ..., M)$   $M$ : Polynomial

Coefficient of the approximated polynomial  $a_j$  (j = 0,...,M)

The relationship among A, a, and f in the matrix is expressed as follows:

$$(\mathbf{A}^T \cdot \mathbf{A}) \cdot \mathbf{a} = \mathbf{A}^T \cdot \mathbf{f}$$

The elements of these matrices are as follows:

$$\{ \mathbf{A}^T \cdot \mathbf{A} \}_{ij} = \sum_{k=-n_L}^{n_R} A_{ki} A_{kj} = \sum_{k=-n_L}^{n_R} k^{i+j}$$

$$\{\mathbf{A}^T \cdot \mathbf{f}\}_j = \sum_{k=-n_L}^{n_R} A_{kj} f_k = \sum_{k=-n_L}^{n_R} k^j f_k$$

Based on the previous expressions, the coefficient Cn is expressed in the following expression by replacing f with en, the unit vector:

$$c_n = \{ (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot (\mathbf{A}^T \cdot \mathbf{e}_n) \}_0 = \sum_{m=0}^M \{ (\mathbf{A}^T \cdot \mathbf{A})^{-1} \}_{0m} n^m \quad (-n_L \le n \le n_R)$$

- 1) sourceX or sourceY is NULL or the number of elements is 0.
- 2) The number of elements is different between sourceX and sourceY.
- 3) smoothingPoints is an even number.
- 4) smoothingPoints is out of the range.
- 5) polynomialOrder is out of the range.
- 6) smoothingPoints 1 < polynomialOrder.
- 7) The interval between sourceX values is not constant.
- 8) The number of sourceX or sourceY elements exceeds 131072.

# 7. Analysis Class

# 7. 1 Member Functions

This section describes the member functions of the Analysis class.

Namespace: ATTAS.Math.Basic32.Analysis

ATTAS. Math. Basic 64. Analysis

### 7. 1. 1 GetVersion

This function returns the arithmetic library version.

# ◆ Syntax

<Excel VBA>

GetVersion()

#### ◆ Return value

<Excel VBA>

Return value type	Description	
String	Arithmetic library version (example: "R1.0").	

# 7. 1. 2 DetectPeak

This function detects peaks of waveform data.

### ◆ Syntax

<Excel VBA>

DetectPeak(source, peakType, smoothingPoints, positiveThreshold,
negativeThreshold, sortType [, peaks])

# 7. Analysis Class

# ◆ Argument

<Excel VBA>

Argument name	I/O	Data type	Description
source	In	Double()	Waveform data array.
peakType	In	Integer	Orientation of peaks to detect. 0: Upward 1: Downward 2: Both directions
smoothingPoints	In	Integer	Smoothing derivation points (odd numbers from 3 to the number of source elements).
positiveThreshold	In	Double	Threshold when detecting positive peaks. Enabled when peakType is 0 or 2.
negativeThreshold	In	Double	Threshold when detecting negative peaks. Enabled when peakType is 1 or 2.
sortType	In	Integer	Method for sorting peak detection results.  0: Does not sort the results.  1: Sorts in ascending order of peak height.  2: Sorts in descending order of peak height.
peaks	In	Integer	Maximum number of peak detections. If 0 is specified, all peaks are detected.

# ◆ Return value

<Excel VBA>

Return value type	Description	
Integer ()	Index array at peak positions in source.	

# ◆ Details

The peak positions are determined according to the following procedure:

1) Smoothing derivation by quadratic polynomial adaptation is performed.

$$Y_i = \sum_{j=1}^{q} ((y_{i+j} - y_{i-j}) \times j)$$
  $(i = 0,1,...,N-1)$ 

 $Y_i$ : Derivation waveform

 $y_i$ : Waveform data

q: Smoothing derivation points

N: Number of elements of the waveform data

- 2) The points, at which the sign of derivation waveform changes from positive to negative, are handled as upward peaks. The points, at which it changes from negative to positive, are handled as downward peaks.
- 3) The detected peak positions are sorted at the peak height according to sortType.
- 4) If the number of detected peaks exceeds the value set to peaks, it is limited to that value.

- 1) The number of source elements is 0.
- 2) peakType is out of the range.
- 3) smoothingPoints is an even number.
- 4) smoothingPoints is out of the range.
- 5) sortType is out of the range.
- 6) peaks is out of the range.

# Appendix 1. Error Messages

An exception is output if an error occurs in the arithmetic library. The following shows the error codes (HResult), descriptions of them, and major causes of each error:

HResult	Description	Main cause of error	
0x80131502	Indicates that the argument is out of the effective range.	A value out of the effective range is specified for the argument.	
0x80131509	Indicates that an error occurred during calculation processing.	Memory shortage. Calculation processing was not performed correctly.	
0x80004003	Indicates that the argument is NULL.	NULL is specified for the argument.	
0x80070057	Indicates that the number of array elements of the argument is invalid.	Array data with 0 elements is specified. The number of X-axis elements does not match that of the Y-axis.	

The following example shows an error code (HResult) when an error occurs and the Excel VBA that displays an error message:

```
Sub main()

' Supplements the exception.
On Error GoTo ErrorHandler

Set axisconv = CreateObject("ATTAS.Math.Basic32.AxisConversion")
Set analy = CreateObject("ATTAS.Math.Basic32.Analysis")
Set calc = CreateObject("ATTAS.Math.Basic32.Calculation")

' Executes calculation.
' ~ (Omitted) ~
Exit Sub

ErrorHandler:
' Displays an error code (HResult).
MsgBox(Err.Number)
' Displays an error message.
MsgBox(Err.Description)
End Sub
```

# Appendix 2. Values of Constants

The following are the values of the constants used for the arithmetic library:

Constant	Value
Pi	3.14159265358979323846
Speed of light [m/s]	2.99792458E+8

# **List of Tables**

Table C-1	Conventions for Command Line and Program Examples	Convention-1
Table C-2	Indicators for Operating the GUI	Convention-1
Table C-3	Indicators for Keyboard Operation	Convention-2
Table C-4	Reference Source Notations	Convention-2