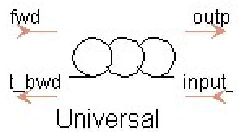
Photonic Modules > Fibers > UniversalFiber

UniversalFiber

Universal Fiber



Purpose

The Universal Fiber model simulates a wideband nonlinear signal transmission in optical fibers with piecewise constant parameters specified for each fiber span individually, taking into account bidirectional signal flow, stimulated and spontaneous Raman and Brillouin scattering, multiple Rayleigh scattering, Kerr nonlinearity, dispersion, PMD effects, local insertion loss, and reflectance at each joint between spans. It is ideal for modeling forward- or reverse-pumped Raman amplifiers, which often form part of the transmission fiber itself. Fiber characteristics measured by an OTDR can be imported into the module as industry-standard data files.

Keywords

Fiber, Bidirectional, Nonlinear, Raman, Brillouin, Stimulated, Spontaneous, Dispersion, Amplifier, SPM, XPM, FWM, PMD, Polarization, Stokes

Inputs

Port	Purpose	Signal/Data Type
input_fwd	input optical signal in the forward direction	Optical Blocks
input_bwd	input optical signal in the backward direction	Optical Blocks

Outputs

Port	Purpose	Signal/Data Type
output_fwd	output optical signal in the forward direction	Optical Blocks
output_bwd	output optical signal in the backward direction	Optical Blocks

Parameters

In this section:

Physical

Name and Description	Unit	Type	Volatile	Value Range	Default Value
NumberOfFiberSpans The number of distinct sections of fiber in the link. Each section can have different characteristics (attenuation, dispersion, dispersion slope, Raman and Brillouin scattering properties, Rayleigh coefficient, nonlinear index, core area, and temperature). It is also possible to specify the reflectance and loss at each joint between spans, using fixed parameters, wavelength-dependent data files, or imported OTDR traces. If data is imported from OTDR trace files, then the number of spans should correspond to the number of discrete events occurring in the OTDR trace.	-	int	yes	[1;∞[[a]	1
Length The length of the fiber span(s). This may be a single value, in which case it is interpreted as the length of every fiber span, or an array of values corresponding to each span in the link. If data is imported from OTDR trace files, this parameter is ignored.	m	floatarray	yes	[0.0;∞[[a]	25.0e3
GroupRefractiveIndex The group refractive index (indices) of the fundamental mode of fiber span(s) at the reference frequency. This may be a single value, in which case it is interpreted as the group index of every fiber span, or an array of values corresponding to each span in the link.	-	floatarray	yes	[0.0;∞[[a]	1.47
AttenuationDescription Defines how the fiber attenuation is specified. Can be set to: AttenuationParameter, in which case a single attenuation value is used; or AttenuationFile, in which case the frequency-dependent attenuation is read from a file.	-	enum	yes	AttenuationParameter, AttenuationFile	AttenuationParameter
Attenuation The attenuation per unit length of the fiber span (s). This may be a single value, in which case it is interpreted as the attenuation of every fiber span, or an array of values corresponding to each span in the link. If data is imported from OTDR trace files, this parameter is ignored. Used when AttenuationDescription is set to AttenuationParameter.	dB/m	floatarray	yes	[0.0;1e-1] [w]	0.2e-3

(context sensitive)					
AttFilename The name of a file containing attenuation data as a function of frequency. If there are multiple spans, this may be a single filename which is applied to all spans, or a set of filenames separated by spaces corresponding to each fiber span. Used when AttenuationDescription is set to AttenuationFile. (context sensitive)	-	inputfilearray	yes	-	-
ReferenceFrequency Reference frequency for fiber dispersion parameters.	Hz	float	yes	]0.0;∞[[a]	193.1e12
DispersionDescription Defines how the fiber dispersion is specified. Can be set to: DispersionParameters, in which case ReferenceFrequency, Dispersion, and DispersionSlope are used; or DispersionFile, in which case the wavelength- dependent dispersion is read from a file.	-	enum	yes	DispersionParameters, DispersionFile	DispersionParameters
Dispersion The dispersion coefficient of the fiber span(s) at the reference frequency. This may be a single value, in which case it is interpreted as the dispersion of every fiber span, or an array of values corresponding to each span in the link. Used when DispersionDescription is set to DispersionParameters. (context sensitive)	s/m^2	floatarray	yes	[-500e-6;500e-6] [w]	16e-6
DispersionSlope The slope of the dispersion characteristic of the fiber span(s) at the reference frequency. This may be a single value, in which case it is interpreted as the dispersion slope of every fiber span, or an array of values corresponding to each span in the link. Used when DispersionDescription is set to DispersionParameters. (context sensitive)	s/m^3	floatarray	yes	[-50e3;50e3] [w]	0.08e3
DispersionFilename The name of a file containing dispersion data as a function of frequency. If there are multiple spans, this may be a single filename which is applied to all spans, or a set of filenames separated by spaces corresponding to each fiber span. Used	-	inputfilearray	yes	-	-

when DispersionDescription is set to DispersionFile. (context sensitive)					
PMDCoefficient The polarization mode dispersion coefficient of the fiber. Used only when PolarizationAnalysis is set to VectorPMD or with parameterized signals.	s/sqrt (m)	float	yes	]-∞;∞[[a]	0.1e-12/31.62
CorrelationLength The correlation length of the fiber birefringence. Used only when PolarizationAnalysis is set to VectorPMD. (context sensitive)	m	float	yes	]0.0;∞[[a]	50.0
RayleighScattering Specifies whether or not Rayleigh backscatter is included in the simulation.	-	enum	yes	Yes, No	No
UseOTDRdata Specifies whether or not the characteristics of a multispan fiber link should be imported from the OTDR files named by the parameters OTDRForwardFilename and/or OTDRBackwardFilename.	-	enum	yes	Yes, No	No

Physical\Nonlinear Effects

Name and Description	Unit	Type	Volatile	Value Range	Default Value
RamanScattering Specifies whether or not Raman scattering is included in the simulation.	-	enum	yes	Yes, No	No
BrillouinScattering Specifies whether or not stimulated Brillouin scattering is included in the simulation.	-	enum	yes	Yes, No	No
Temperature The temperature of the fiber span(s), used for calculation of the spontaneous Raman and Brillouin emission. This may be a single value, in which case it is interpreted as the temperature of every fiber span, or an array of values corresponding to each span in the link. (context sensitive)	K	floatarray	yes	[0.0;10000.0] [a]	300.0
NonlinearDescription Defines how the nonlinearity of the fiber is specified. Can be set to: NonlinearIndexParameter, in which case a nonlinear coefficient calculated from the NonLinearIndex is used; or NonlinearFile, in which case the nonlinear coefficient is read from file defined by NonlinearFilename.	-	enum	yes	NonlinearIndexParameter, NonlinearFile	NonlinearIndexParameter
NonLinearIndex Nonlinear refractive index measured with constant linear polarization. For fibers with randomly varying birefringence and PolarizationAnalysis=Scalar, the nonlinear terms should be	m^2/W	floatarray	yes	[-1e-15;1e-15] [W]	2.6e-20

corrected using the parameter NonlinearAdjustmentFactors. For PolarizationAnalysis=VectorPMD, the nonlinear terms are automatically corrected. This parameter may be a single value, in which case it is interpreted as the nonlinear index of every fiber span, or an array of values corresponding to each span in the link. Used only when NonlinearDescription is set to NonlinearIndexParameter. (context sensitive)					
NonLinearFilename The name of a file containing the nonlinear coefficient as a function of frequency. The nonlinear coefficient is defined as gamma(f) = 2 pi n2(f) f / c / Aeff(f), where the nonlinear index n2 is measured with constant linear polarization. For fibers with randomly varying birefringence and PolarizationAnalysis=Scalar, the nonlinear terms should be corrected using the parameter NonlinearAdjustmentFactors. For PolarizationAnalysis=VectorPMD, the nonlinear terms are automatically corrected. If there are multiple spans, this parameter may be a single filename which is applied to all spans, or a set of filenames separated by spaces corresponding to each fiber span. Used only when NonlinearDescription is set to NonlinearFile. (context sensitive)	-	inputfilearray	yes	-	-
CoreAreaDescription Defines whether the core area or overlap integral is specified. May be set to: CoreAreaParameter, in which case the core area and overlap integrals are assumed to be frequency-independent; CoreAreaFile, in which case the frequency-dependent effective core area is read from the file defined by CoreAreaFilename; OverlapIntegralFile, in which case the model uses overlap integrals, which are specified in the file defined by OverlapIntegralFilename.	-	enum	yes	CoreAreaParameter, CoreAreaFile, OverlapIntegralFile	CoreAreaParameter
CoreArea The effective core area of the fiber span(s). This may be a single value, in which case it is interpreted as the core area of every fiber span, or an array of values corresponding to each span in the link. Used only when CoreAreaDescription is set to CoreAreaParameter. (context sensitive)	m^2	floatarray	yes	]0.0;∞[[a]	80.0e-12
CoreAreaFilename The name of a file containing effective core area data as a function of frequency. If there are multiple spans, this may be a single filename which is applied to all spans, or a set of filenames separated by spaces corresponding to each fiber span. Used only when CoreAreaDescription is set to	-	inputfilearray	yes	-	-

CoreAreaFile. (context sensitive)					
OverlapIntegralFilename The name of a file containing the overlap integral as a function of two frequencies. If there are multiple spans, this may be a single filename which is applied to all spans, or a set of filenames separated by spaces corresponding to each fiber span. Used when CoreAreaDescription is set to OverlapIntegralFile. (context sensitive)	-	inputfilearray	yes	-	-
RamanFraction A coefficient defining the fractional contribution of the delayed Raman response to the entire nonlinearity of the fiber span(s). This may be a single value, in which case it is applied to every fiber span, or it may be an array of values corresponding to each span in the link. If RamanScattering = No, it is assumed that the specified NonLinearIndex is purely electronic; therefore, a zero value of RamanFraction will be used in calculations. (context sensitive)	-	floatarray	yes	[0.0;1.0][a]	0.17
SPM_EC Specifies whether or not to take into account the electronic contribution to SPM. (context sensitive)	-	enum	yes	No, Yes	Yes
XPM_EC Specifies whether or not to take into account the electronic contribution to XPM. (context sensitive)	-	enum	yes	No, Yes	Yes
XPM_MC Specifies whether or not to take into account the molecular contribution to XPM. (context sensitive)	-	enum	yes	No, Yes	Yes
IntraBandRaman Specifies whether or not to take into account intraband Raman effects. (context sensitive)	-	enum	yes	No, Yes	Yes
InterBandRaman Specifies whether or not to take into account interband Raman effects. (context sensitive)	-	enum	yes	No, Yes	Yes
FWM Defines whether to consider FWM interactions in power analysis equations or not. Used only when PolarizationAnalysis = Scalar. (context sensitive)	-	enum	yes	No, Yes	No
HigherOrderNLEffects Specifies whether or not to include higher-order nonlinear effects during the field analysis. (context sensitive)	-	enum	yes	Yes, No	No
NonlinearAdjustmentFactors Five (seven) adjustment factors that can be used to control the strength of nonlinear interactions between copropagating bands during the field analysis: (1) SPM_EC term,	-	floatarray	yes	[0.0;10.0][a]	(8./9.) 1.0 (2./3.) 1.0 (1./2.) 1.0 1.0

(2) IntraBandRaman term, (3) XPM_EC term, (4) XPM_MC term, (5) InterBandRaman term, (6) FWM_Kerr term, and (7) FWM_Raman term. (6) and (7) are used if FWM = Yes. Used only when PolarizationAnalysis = Scalar. (context sensitive)					

Physical\Nonlinear Effects\Raman Scattering Parameters

Name and Description	Unit	Type	Volatile	Value Range	Default Value
RamanFilename The name of a file containing the Raman gain as a function of frequency difference. If there are multiple spans, this may be a single filename which is applied to all spans, or a set of filenames separated by spaces corresponding to each fiber span. (context sensitive)	-	inputfilearray	yes	-	$BNROOT/data/UFM/RamanGain.dat
SpontaneousRamanScattering Specifies whether or not to generate spontaneous Raman scattering. If SpontaneousRamanScattering = Yes, noise bins are generated as specified by parameters NoiseBandwidth, NoiseCenterFrequency, and FreqResolutionNB. (context sensitive)	-	enum	yes	No, Yes	No
RamanAdjustmentFactors An array of 1, 2, 5, or 9 values specifying adjustment factors for stimulated (ST) and spontaneous (SP) Raman scattering of signals (S), noise (N), and distortions (D). If one factor is specified, it is used to scale the interaction between all kinds of signals. If two factors are specified, the 1st one corresponds to the stimulated, and the second to spontaneous Raman interactions. If five factors are entered, they correspond to 1) S-S ST 2) S-N,S-D ST 3) N-N,N-D,D-D ST 4) S-N SP 5) N-N,D-N SP interactions. If nine factors are entered, they correspond to 1) S-S ST 2) S-N ST 3) S-D ST 4) N-N ST 5) N-D ST 6) D-D ST 7) S-N SP 8) N-N SP 9) D-N SP interactions. This parameter is used only when PolarizationAnalysis is set to Scalar. See the manual for more details. (context sensitive)	-	floatarray	yes	[0.0;10.0] w	0.5

Physical\Nonlinear Effects\Brillouin Scattering Parameters

Name and Description	Unit	Type	Volatile	Value Range	Default Value
SBSBandwidth The effective bandwidth of the Brillouin line shape. (context sensitive)	Hz	float	yes	]0.0;∞[[a]	100e6

SBSStokesShift The stimulated Brillouin scattering frequency shift. The value will be quantized to 1/TimeWindow frequency grid. (context sensitive)	Hz	float	yes	]0.0;∞[[a]	11e9
SBSDescription Defines how the Brillouin gain profile is defined. May be set to: SBSGainParameter, in which case the Brillouin gain is assumed to be frequency-independent and defined by the SBSGain parameter; or SBSGainFile, in which case the frequency dependent Brillouin gain is read from a file defined by the parameter SBSGainFilename. (context sensitive)	-	enum	yes	SBSGainParameter, SBSGainFile	SBSGainParameter
SBSGain Stimulated Brillouin scattering gain coefficient. This may be a single value, in which case it is interpreted as a gain coefficient of every fiber span, or an array of values corresponding to each span in the link. Used when SBSDescription is set to SBSGainParameter. (context sensitive)	m/W	floatarray	yes	]0.0;∞[[a]	4.6e-11
SBSGainFilename The name of a file which contains the frequency-dependent SBS gain. Used when SBSDescription is set to SBSGainFile. If there are multiple spans, this may be a single filename which is applied to all spans, or a set of filenames separated by spaces corresponding to each fiber span. (context sensitive)	-	inputfilearray	yes	-	-
SBSAdjustmentFactor Polarization adjustment factors for SBS effect. If one factor is specified, it is used to scale stimulated and spontaneous Brillouin terms. If two factors are specified, the first corresponds to the stimulated, and the second to spontaneous Brillouin interaction. Used only when PolarizationAnalysis is set to Scalar. (context sensitive)	-	floatarray	yes	[0.0;1000.0][a]	0.5

Physical\Rayleigh Scattering Parameters

Name and Description	Unit	Type	Volatile	Value Range	Default Value
BackscatterDescription	-	enum	yes	Linear, OTDR	OTDR

Defines whether the backscatter is entered as the proportion of power reflected per meter (Linear), or as the measured reflection of a pulse with defined width (OTDR). (context sensitive)					
EffectiveGroupIndex The effective group refractive index used to convert the OTDR-measured back reflection for a defined pulse width into a proportion per meter. Used only when BackscatterDescription is set to OTDR. (context sensitive)	-	float	yes	[1.0;100.0][a]	1.47
OTDRPulseWidth The width of the pulse used to measure the backreflection coefficient in an OTDR. This is used to convert to a proportion per meter. Used only when BackscatterDescription is set to OTDR. (context sensitive)	s	float	yes	]0.0;100.0][a]	1e-9
RayleighDescription Defines how the fiber Rayleigh backscattering coefficient is specified. Can be set to: RayleighCoefficient, in which case a single wavelength-independent coefficient is used; or RayleighFile, in which case the wavelength-dependent Rayleigh backscattering coefficient is read from a file. (context sensitive)	-	enum	yes	RayleighCoefficient, RayleighFile	RayleighCoefficient
RayleighBackscatterCoefficient If BackscatterDescription = Linear, this parameter specifies backscattering coefficient as the proportion of power reflected per meter (1/m units). If BackscatterDescription = OTDR, this parameter specifies backscattering coefficient as the measured reflection of a pulse with defined width (dB units). This may be a single value, in which case it is interpreted as the backscatter coefficient of every fiber span, or an array of values corresponding to each span in the link. If data is imported from OTDR trace files, this parameter is ignored. Used only when RayleighDescription is set to RayleighCoefficient. (context sensitive)	dB | 1/m	floatarray	yes	]-∞;∞[[a]	-80.0
RayleighFilename The name of a file containing Rayleigh backscatter data as a function of frequency. If there are multiple spans, this may be a single filename which is applied to all spans, or a set of filenames separated by spaces corresponding to each fiber span. Used only when RayleighDescription is set to RayleighFile. (context sensitive)	-	inputfilearray	yes	-	-

Physical\Event Loss and Reflectance

Name and Description	Unit	Type	Volatile	Value Range	Default Value
EventLossDescription Defines how the fiber event loss is specified. Can be set to: EventLossParameter, in which case a single event loss value is used; or EventLossFile, in	-	enum	yes	EventLossParameter, EventLossFile	EventLossParameter

which case the frequency-dependent event loss is read from a file.					
EventLoss The losses incurred at event locations (the fiber ends and joins between fiber spans). This may be a single value, in which case it is interpreted as the loss at all event locations, or an array of values corresponding to each event location in the link. If an array is provided, then the number of values must be equal to the number of fiber spans plus one. Used when EventLossDescription is set to EventLossParameter. (context sensitive)	dB	floatarray	yes	[0.0;∞[[a]	0.0
EventLossFilename The name of a file containing the losses incurred at event locations as a function of frequency or wavelength. This may be a single filename which is applied to all event locations, or a set of filenames separated by spaces corresponding to each event location. If multiple filenames are provided, then the number of files must be equal to the number of fiber spans plus one. Used when EventLossDescription is set to EventLossFile. (context sensitive)	-	inputfilearray	yes	-	-
EventReflectanceDescription Defines how the fiber event reflectance is specified. Can be set to: EventReflectanceParameter, in which case a single event reflectance value is used; or EventReflectanceFile, in which case the frequency-dependent event reflectance is read from a file.	-	enum	yes	EventReflectanceParameter, EventReflectanceFile	EventReflectanceParameter
EventReflectance The strength of reflections (return loss) occurring at event locations (the fiber ends and joins between fiber spans). This may be a single value, in which case it is interpreted as the return loss at all event locations, or an array of values corresponding to each event location in the link. If an array is provided, then the number of values must be equal to the number of fiber spans plus one. Used when Event ReflectanceDescription is set to EventReflectanceParameter. (context sensitive)	dB	floatarray	yes	[-400;0.0[[a]	-400.0
EventReflectFilename The name of a file containing the return losses occurring at event locations as a function of frequency or wavelength. This may be a single filename which is applied to all event locations, or a set of filenames separated by spaces corresponding to each event location. If multiple filenames are provided, then the number of files must be equal to the number of fiber spans plus one. Used when EventReflectanceDescription is set to EventReflectanceFile.	-	inputfilearray	yes	-	-

(context sensitive)						

Physical\OTDR Files

Name and Description	Unit	Type	Volatile	Value Range	Default Value
`OTDRForwardFilename` The name of a file containing measured OTDR data for the forward direction of the fiber link, in SR-4731 format. Multiple space-separated filenames may be provided, in which case each file should contain data corresponding to a different measurement wavelength. The parameter NumberOfFiberSpans must be one less than the number of key events identified in the OTDR file. (context sensitive)	-	inputfilearray	yes	-	-
`OTDRBackwardFilename` The name of a file containing measured OTDR data for the backward direction of the fiber link, in SR-4731 format. Multiple space-separated filenames may be provided, in which case each file should contain data corresponding to a different measurement wavelength. The parameter NumberOfFiberSpans must be one less than the number of key events identified in the OTDR file. (context sensitive)	-	inputfilearray	yes	-	-
`OTDRLogFilename` Name of a file in which to store the fiber parameters retrieved from the OTDR file. (context sensitive)	-	outputfile	yes	-	-

Numerical\Spectral Discretization

Name and Description	Unit	Type	Volatile	Value Range	Default Value
`FreqResolutionNB` Frequency grid interval used for the computation of the noise spectral shape. This parameter specifies the width of the noise bins (NB) generated due to spontaneous Raman scattering.	Hz	float	yes	]0.0;∞[[a]	100e9
`FreqResolutionSFB` Frequency grid interval used for the computation of the signal spectral shape. Input sampled frequency bands (SFB) are replaced with parameterized signals on each grid interval.	Hz	float	yes	]0.0;∞[[a]	100e9
`GridReferenceFrequency` Used for alignment of the frequencies of noise bins and sampled band discretization points. The center frequency of one of the generated noise bins is equal to this parameter.	Hz	float	yes	[0.0;∞][a]	193.1e12
`SpectralDiscretizerDescription` Defines if the frequencies of discretized sampled bands are equal to the center frequencies of the corresponding FreqResolutionSFB intervals (FixedFrequencies) or calculated as average weighted frequencies over all samples within the FreqResolutionSFB interval.	-	enum	yes	FixedFrequencies, AverageWeighted	FixedFrequencies
`SBSThresholdFactor` For input sampled bands, Brillouin scattering is considered for samples with power larger than the 1/BrillouinThresholdFactor *	-	float	yes	[1.0;∞][a]	10.0

power of the sample with the highest power. (context sensitive)					
FWMThreshold Threshold for FWM distortions. The distortions with the power lower than this threshold are discarded. (context sensitive)	(dBm)	float	yes	]-∞;∞[[a]	-30.0
FWMPowerSeed The initial power (seed) for generated FWM distortions. (context sensitive)	W	float	yes	]0.0;∞[[a]	1e-10

Numerical

Name and Description	Unit	Type	Volatile	Value Range	Default Value
NoiseCenterFrequency Center frequency of the noise bandwidth. (context sensitive)	Hz	float	yes	[0.0;∞] [a]	193.1e12
NoiseBandwidth Bandwidth around NoiseCenterFrequency where excess noise due to spontaneous Raman scattering is considered. (context sensitive)	Hz	float	yes	[0.0;∞] [a]	10e12

Numerical\Boundary Value Problem Solver

Name and Description	Unit	Type	Volatile	Value Range	Default Value
AccuracyGoal Target relative accuracy for the solution of the two-point boundary value problem during bidirectional power analysis. The accuracy estimate is determined by comparing numerical solutions while reducing the step size. Note that the step size cannot be reduced below MinimumStepSize, which may prevent the AccuracyGoal from being achieved.	dB	float	yes	[0.0;∞[[a]	0.01
MaximumIterations Maximum number of iterations used to solve the two-point boundary value problem during bidirectional power analysis.	-	int	yes	[1;100000] [a]	1000
IterationAccuracyFactor A factor controlling the target relative accuracy when performing internal iterations with constant step size to solve the boundary problem for the bidirectional power analysis. In order to achieve the desired overall target accuracy specified by the parameter AccuracyGoal, the internal iterative procedure must achieve a higher accuracy. This parameter specifies the factor by which the accuracy of the iterative procedure should exceed the overall AccuracyGoal.	-	float	yes	[2.0;10000.0] [a]	10.0
InitialStepSize Step size of the initial grid used for the bidirectional power analysis. This may be a single value, in which case it is interpreted as the initial step size for every fiber span, or an array of values corresponding to each span in the link.	m	floatarray	yes	]0.0;∞[[a]	1000.0
MinimumStepSize Minimum acceptable step size of the grid used for the bidirectional power analysis. The algorithm uses a variable grid for power calculations, which may be reduced in order to achieve the desired AccuracyGoal. This parameter sets the	m	float	yes	]0.0;∞[[a]	1.0

smallest step size that can be used. In the case of a multiple span fiber link, the algorithm is not allowed to use a step size less than the MinimumStepSize on any span.					
StabilizationFactor Stabilization factor for the two-point boundary value solver algorithm used in power analysis. Setting StabilizationFactor=1.0 employs no stabilization and usually gives maximum computation speed. Smaller values provide better stability, usually at the expense of computation speed.	-	float	yes	]0.0;1.0][a]	0.5

Numerical\Split-Step Parameters

Name and Description	Unit	Type	Volatile	Value Range	Default Value
StepSelectionMethod Defines the method for selecting the step size in the field analysis. When LocalErrorMethod is selected, the step size is governed by TargetLocalError parameter. If NonlinearPhaseChange is selected, the step size is specified by MaxPhaseChange parameter. For ConstantStep setting, the fixed step size equal to MaxStepWidth is used in simulation. (context sensitive)	-	enum	yes	LocalErrorMethod, NonlinearPhaseChange, ConstantStep	NonlinearPhaseChange
SplitStepType Switches between the symmetric and asymmetric split-step Fourier methods for the field analysis. (context sensitive)	-	enum	yes	Symmetric, Asymmetric	Symmetric
MaxStepWidth Maximum step width allowed in the split-step Fourier method used in the field analysis. (context sensitive)	m	float	yes	]0.0;∞[[a]	1.0e3
MinStepWidth The minimum allowed step width in the split-step Fourier method used in the field analysis. (context sensitive)	m	float	yes	[0.0;∞[[a]	0.0
InitialStepWidth Initial step width for local error method in the split-step Fourier algorithm used in the field analysis. (context sensitive)	m	float	yes	]0.0;∞[[a]	1.0e3
TargetLocalError Maximum tolerable local error in the split-step Fourier method used in the field analysis. The step width is chosen such that the local error on a single simulation step is within [TargetLocalError/2; 2*TargetLocalError].	-	float	yes	]0;∞[[a]	1.0e-5

Smaller values of TargetLocalError result in smaller steps, which provide better accuracy but require longer computation time. The upper and lower limits of the step width are set by the parameters MaxStepWidth and MinStepWidth, respectively. (context sensitive)					
MaxPhaseChange Maximum tolerable phase change in the split-step Fourier method used in the field analysis. The step width is chosen such that the phase of the optical field in the nonlinear step does not change by more than MaxPhaseChange over the step. Smaller values of MaxPhaseChange result in smaller steps, which provide better accuracy but require longer computation time. The upper and lower limits of the step width are set by the parameters MaxStepWidth and MinStepWidth, respectively. (context sensitive)	deg	float	yes	]0;∞[[a]	0.05

Numerical\PMD Coarse Step Parameters

Name and Description	Unit	Type	Volatile	Value Range	Default Value
MeanStepWidth Mean step width about which the scattering step widths are distributed when performing polarization scattering. Used only when PolarizationAnalysis is set to VectorPMD. (context sensitive)	m	float	yes	]0.0;∞[[a]	1.0e3
WidthDeviation Standard deviation of the polarization scattering step width around the MeanStepWidth. Used only when PolarizationAnalysis is set to VectorPMD. (context sensitive)	m	float	yes	[0.0;∞] [a]	100.0

Enhanced

Name and Description	Unit	Type	Volatile	Value Range	Default Value
PolarizationAnalysis This switch defines how the polarization is handled within the module. If set to Scalar, the polarization properties of nonlinear effects are included only by means of adjustment factors; if set to VectorPDM, a vectorial PMD model is applied and polarization dependence of nonlinear effects is considered.	-	enum	yes	Scalar, VectorPMD	Scalar
BidirectionalAnalysis When signals propagate in the fiber in only one direction, setting this parameter to Simplified can significantly speed up the power analysis. However, if signals propagate in both directions, the interactions between counterpropagating	-	enum	yes	Simplified, Full	Full

signals are not considered in the Simplified mode. It means that the effects of Raman scattering of counterpropagating signals, Rayleigh scattering, Brillouin scattering, and bidirectional noise generation cannot be included if BidirectionalAnalysis = Simplified. If this parameter is set to Full, all interactions between counterpropagating signals are considered and all the effects can be included.					
`FieldAnalysis` Specifies whether or not to perform field analysis in forward and/or backward directions.	-	enum	yes	No, Forward, Backward, Bidirectional	Forward
`SkipFieldAnalysis` When this module is used in iterations, the field analysis is skipped on the first iterations whose number is specified by this parameter. The field analysis is performed only starting from the iteration with number (SkipFieldAnalysis+1). This option can be useful if several UniversalFiber modules are connected to each other. (context sensitive)	-	int	yes	[0;10000][a]	0
`NoiseBinInteractions` If this parameter is set to No, then the Raman power transfer from noise bins to signals and other noise bins is neglected. This approximation is usually acceptable when the signal power is much greater than the noise power, and can speed up the computation significantly.	-	enum	yes	No, Yes	Yes
`DistortionInteractions` If this parameter is set to No, then the Raman power transfer from distortions to signals and other distortions is neglected. This approximation is usually acceptable when the signal power is much greater than the power of distortions, and can speed up the computation significantly.	-	enum	yes	No, Yes	Yes
`NoisePolarization` This switch defines the polarization of the noise generated in the fiber. Please note that for Unpolarized mode the total amount of spontaneous Raman noise is two times higher than for NoisePolarization=X or Y because in this case the noise is generated into two modes. This parameter is effective only for PolarizationAnalysis=Scalar. (context sensitive)	-	enum	yes	Unpolarized, X, Y	Unpolarized
`DistortionPolarization` This switch defines the polarization of the Rayleigh and Brillouin distortions generated in the fiber. Please note that for Unpolarized mode the total amount of spontaneous Brillouin noise is two times higher than for NoisePolarization = X or Y because in this case the noise is generated into two modes. This parameter is effective only for PolarizationAnalysis=Scalar. (context sensitive)	-	enum	yes	Unpolarized, X, Y	Unpolarized
`ConserveMemory` Specifies whether to run the field analysis using memory-conserving or memory-intensive algorithms. Note that on systems with adequate available RAM, the memory-intensive algorithms are faster. (context sensitive)	-	enum	yes	No, Yes	No
`NoiseRandomNumberSeed` The random seed lookup indexes to be used for incorporating noise bins into sampled bands (if the global parameter InBandNoiseBins=Off). This parameter is an array that may contain one or two values. If two values are provided, then the first is the lookup index used for	-	intarray	no	[0;10000][a]	0

forward-propagating noise, and the second is the lookup index for backward-propagating noise. If only one value is provided, then it is used for both forward- and backward-propagating noise. In either case, a value of zero causes a unique seed to be automatically selected.					
Active Defines if the module is active or not.	-	enum	yes	On, Off	On

Enhanced\PMD Auxiliary Parameters

Name and Description	Unit	Type	Volatile	Value Range	Default Value
NonlinearPMDBandwidth The bandwidth around a signal in which nonlinear polarization-dependent interactions with other signals are considered. The nonlinear polarization-dependent interactions with the signals outside this range are ignored because the polarization of these signals changes very quickly due to PMD and its averaged nonlinear contribution is negligible. (context sensitive)	Hz	float	no	[0.0;∞][a]	30.0e12
WidthRandomNumberSeed Random seed lookup index for the polarization scattering step width. A value of zero uses an automatic unique seed. (context sensitive)	-	int	no	[0;10000][a]	0
ScatteringRandomNumberSeed Random seed lookup index for random scattering matrices. A value of zero uses an automatic unique seed. (context sensitive)	-	int	no	[0;10000][a]	0
FreezeWidth Freezes the step width for a given number of calls. Used to isolate numerical from physical effects, for example, for Jones Matrix analysis. This parameter is effective only for PolarizationAnalysis=VectorPMD. (context sensitive)	-	int	no	[0;10000][a]	0
FreezeScattering Freezes scattering for a given number of calls. Used to isolate numerical from physical effects, for example, for Jones Matrix analysis. This parameter is effective only for PolarizationAnalysis=VectorPMD. (context sensitive)	-	int	no	[0;10000][a]	0
CorrelationScatteringFactor Degree of correlation between successive polarization scattering sections. (context sensitive)	-	float	no	[0.0;1.0[[a]	0.0
BirefringenceProfile Option to store or retrieve the scattering steps and rotation axes to or from file. In Retrieve mode, data is read from the file specified by the parameter BirefringenceInputFilename. In Store mode, data is written to the file specified by the parameter BirefringenceOutputFilename. (context sensitive)	-	enum	yes	None, Store, Retrieve	None

BirefringenceFileDataType Defines whether the rotating matrices or rotation axes are stored to or retrieved from the file. (context sensitive)	-	enum	yes	RotationMatrices, RotationAxes	RotationMatrices
BirefringenceInputFilename Name of a file from which to retrieve rotation axes and scattering positions. (context sensitive)	-	inputfile	yes	-	-
BirefringenceOutputFilename Name of a file in which to store rotation axes and scattering positions. (context sensitive)	-	outputfile	yes	-	-
LikelihoodRatiosFilename Name of a file in which to store the likelihood ratios. (context sensitive)	-	outputfile	yes	-	-

Visualization

Name and Description	Unit	Type	Volatile	Value Range	Default Value
VisualizationMode Defines whether simulation result is displayed after simulation automatically, or is stored and can be visualized later by demand, or is not generated at all.	-	enum	yes	StartAutomatically, StartOnDemand, None	None
Title Title used to identify internal visualizer plots. (context sensitive)	-	string	yes	-	-
SaveToFile Defines whether selected signal distributions are saved to the file.	-	enum	yes	On, Off	Off
LogFilename The name of a file in which to save the power/Stokes vector distributions along the fiber. (context sensitive)	-	outputfile	yes	-	-
PlotFrequencyRange Frequency range(s) of the signals to be visualized on the plots or saved to file. The power distributions will be visualized or saved to file only for signals with frequencies lying in the specified range(s). One or several frequency ranges can be specified, like (190e12, 191e12) (193e12, 194e12). (context sensitive)	Hz	complexarray	yes	[0.0;∞[[a]	(100e12, 1000e12)
PowerPlotPoints Specifies the number of points to be used for visualization of the power distributions along the fiber. Note, that at the fiber ends and at the event points the signal is depicted as two points with the same abscissa and the ordinates equal to signal power before and after event. Thus, the final number of points is greater that specified by parameter PowerPlotPoints. (context sensitive)	-	int	yes	[2;∞] [a]	101
SignalDirection Defines which signal internal	-	enum	yes	Forward, Backward, Both	Both

distributions are visualized or saved to the file. (context sensitive)					
SeparateWindows Defines whether power axial distributions of parameterized signals, distortions, and noise bins will be shown in separate windows. (context sensitive)	-	enum	yes	On, Off	Off
Signals Defines whether to select internal distribution of signals inside the fiber for visualization or saving to file. For the Block signals, the power distributions of corresponding parameterized signals derived during discretization in power analysis are selected. (context sensitive)	-	enum	yes	On, Off	Off
NoiseBins Defines whether to select internal distribution of noise bins inside the fiber for visualization or saving to file. If set to TotalPower, the total power of all noise bins is visualized or saved to file. If set to IndividualPowers, the power distribution of each noise bin is visualized or saved to file. (context sensitive)	-	enum	yes	TotalPower, IndividualPowers, Off	Off
Distortions Defines whether to select internal distribution of distortions inside the fiber for visualization or saving to file. After setting to On, the individual types of distortions can be selected. (context sensitive)	-	enum	yes	On, Off	Off
CrosstalkDistortions Defines whether to select internal distribution of CrosstalkDistortions inside the fiber for visualization or saving to file. (context sensitive)	-	enum	yes	On, Off	On
RayleighDistortions Defines whether to select internal distribution of RayleighDistortions inside the fiber for visualization or saving to file. (context sensitive)	-	enum	yes	On, Off	On
BrillouinDistortions Defines whether to select internal distribution of BrillouinDistortions inside the fiber for visualization or saving to file. (context sensitive)	-	enum	yes	On, Off	On
FWMDistortions Defines whether to select internal distribution of FWMDistortions inside the fiber for visualization or saving to file. (context sensitive)	-	enum	yes	On, Off	On
GenericDistortions Defines whether to select internal distribution of GenericDistortions inside the fiber for visualization or saving to file. (context sensitive)	-	enum	yes	On, Off	On

`StokesVector` Defines whether to select internal distribution of normalized or nonnormalized Stokes vector components for visualization or saving to file. Stokes vector components are selected for the signals for which the power distribution is selected. (context sensitive)	-	enum	yes	On, Off, Normalized	Off
`ConvergencePower` Defines whether to visualize the convergence of the boundary value problem solver for the power analysis. Convergence is the relative error versus iteration number for the power (Stokes vector component S0). The worst error among all signals is selected. (context sensitive)	-	enum	yes	On, Off	Off
`ConvergenceStokes` Defines whether to visualize the convergence of the boundary value problem solver for the power analysis. ConvergenceStokes is the worst relative error versus iteration number among the Stokes vector components S1, S2 and S3. The worst error among all signals is selected. (context sensitive)	-	enum	yes	On, Off	Off
`ConvergenceGrid` Defines whether to visualize the convergence plot for the grids with different step size. Each point on the plot defines the relative error between the solutions at two consecutive grids. (context sensitive)	-	enum	yes	On, Off	Off

Description

The Universal Fiber Model is so named because it incorporates almost all phenomena in a transmission fiber, even if the fiber is operated with forward and backward signals and Raman pumps. It is not intended to model doped fibers, however.

> **Note:** A two-port version of *UniversalFiber* called *UniversalFiberFwd* is available. This documentation is for both versions.

The characteristics of the fiber are described by a large set of parameters. However, these parameters have been made context-sensitive so that if a particular effect is of no interest, the parameters can be simply ignored and do not appear in the parameter list.

Context-Sensitive Parameters

Most features and parameter choices are selected using enumerated parameters that act as switches between different calculation methods, or ways of representing the fiber's properties. The following list contains the switches which control the global behavior of the module.

- **`PolarizationAnalysis`** defines how the polarization is handled within the module. It can be set to:
  - `Scalar`: the polarization properties of nonlinear terms are included by means of adjustment factors. All PMD terms are ignored.
  - `VectorPMD`: a vectorial PMD model is applied. Linear PMD effect is considered. Nonlinear terms are included in averaged form, the nonlinear PMD terms with zero means are excluded.

  A detailed description of these two regimes is given in Analysis Techniques.
- **`FieldAnalysis`** specifies whether or not to perform field analysis in specified direction. It can be set to:
  - `No`: field analysis is not carried out in any direction. Signals are simply filtered and their power levels changed.
  - `Forward`: field analysis is performed in forward direction.

  + `Backward`: field analysis is performed in backward direction.
  + `Both`: field analysis is performed in both directions.
- **`BidirectionalAnalysis`** specifies whether:
  + `Full`: the equations describe the bidirectional power interactions of the signals.
  + `Simplified`: the signals propagated in different directions do not interact with each other.

  For a more detailed description of this switch, refer to Power Analysis.

Three parameters **`RamanScattering`**, **`BrillouinScattering`**, and **`RayleighScattering`** switch on and off corresponding to physical effects:
- The **`RamanScattering`** parameter "cuts off" all the effects related to delayed molecular response, as well as spontaneous Raman scattering.
- **`RamanScattering`** is also controlled by **`SpontaneousRamanScattering`**, which specifies whether to account for spontaneous Raman scattering.
- If **`SpontaneousRamanScattering`** = `Yes`, noise bins are generated as specified by parameters **`NoiseBandwidth`**, **`NoiseCenterFrequency`**, and **`FreqResolutionNB`**.

In the equations for power and field analysis, the different terms can be included or excluded by switching the corresponding parameters **`SPM_EC`**, **`XPM_EC`**, **`XPM_MC`**, **`IntraBandRaman`**, **`InterBandRaman`**. The last three effects are disabled if **`RamanScattering`** is set to `No`.

The switches such as **`AttenuationDescription`** define how the physical quantity is specified: by means of frequency-independent parameter or by the file. For a more detailed description, refer to Power Analysis and field analysis.

`UseOTDRdata` specifies whether or not the characteristics of a multispan fiber link should be imported from the OTDR files named by the parameters **`OTDRForwardFilename`** and/or **`OTDRBackwardFilename`**. For a more detailed specification of OTDR features, refer to Appendix.

`SplitStepType` switches between the symmetric (it is set to `Symmetric`) and asymmetric (`Asymmetric`) split-step method for field analysis.

Concatenating UniversalFiber Modules

In many cases, several *UniversalFiber* modules are connected to each other at both forward and backward inputs. As the scheduler does not start simulation when there is no signal at one of the fiber inputs, wire delays should be set at the corresponding inputs. The number of runs must be set to achieve global convergence of the schematic. The number must be greater than the maximum specified logical delay. Power analysis works much faster than field analysis, so in such iterations it is not necessary to perform field analysis until global convergence is achieved. The **`SkipFieldAnalysis`** parameter defines the number of runs (or input blocks) in which field analysis should be skipped. It doesn't start until the iteration with the number (**`SkipFieldAnalysis`** + 1). For example, if **`SkipFieldAnalysis`** is set to 4, field analysis starts at the fifth iteration.

Overview of Modeled Effects

In this section:

The transmission effects are grouped into three general categories, each containing some subclasses (refer to Table 1).

Table 1 ***Effects included in UniversalFiber model***

Dispersion Properties	Optical Nonlinearities	Polarization Properties
chromatic dispersion	self-phase modulation cross-phase modulation four-wave mixing (modulation instability) (soliton formation) stimulated and spontaneous Raman scattering stimulated and spontaneous Brillouin scattering	polarization mode dispersion polarization dependent gain/loss

All the nonlinear effects can be generally categorized as either scattering effects:
- stimulated Raman scattering

- stimulated Brillouin scattering,

or effects related to the intensity dependence of the refractive index (Kerr effect):

- self-phase modulation
- cross-phase modulation
- modulation instability
- soliton formation
- four-wave mixing.

These nonlinear effects interact with dispersion, polarization effects and linear scattering to produce complex degradations, or beneficial interactions. For example:

- Four-wave mixing depends strongly on the dispersion of the fiber
- Self-phase modulation can support soliton propagation in the presence of dispersion
- Polarization mode dispersion can be reduced by nonlinear interactions.

Thus, a fiber model is required that includes nonlinear and linear phenomena, so that detrimental and beneficial interactions between these phenomena are predicted correctly.

Attenuation

Fiber losses reduce the power of the signal propagated in the fiber. The most important factors which contribute to overall losses are material absorption of the fused silica, impurities, and Rayleigh scattering. Wavelength-independent fiber losses can be entered as **`Attenuation`** coefficient. For wide-band systems, spectrally dependent fiber attenuation is specified by means of a file, **`AttFileName`**.

Dispersion

This model covers single-mode fibers, so **`Dispersion`** means chromatic or group velocity dispersion for the fundamental mode. Different spectral components propagate with different group velocities, which causes pulse spreading along the fiber. Note that chromatic dispersion is a collective effect of material dispersion, waveguide dispersion and profile dispersion.

Raman Scattering

Stimulated Raman scattering is an inelastic process when light is scattered by material scattering centers, which causes vibrational excitation of the molecules. As a result, lower-energy photons are created from higher-energy photons, due to the energy lost to the material: in effect, shorter wavelength signals are attenuated and longer wavelength signals are amplified. This effect has an important application for WDM systems — relatively high amplifications may be achieved within the transmission fiber. Raman scattering is present at all optical frequencies. The amplification range using a single pump wavelength is extremely broadband — the half-gain bandwidth is about 6 THz at 1.5 μm. Even broader amplifiers can be made with multiple pumps over a range of wavelengths.

The main advantages of using SRS amplification:

- Raman amplification is low-noise compared with EDFAs placed at the ends of links, because it amplifies within the transmission fiber
- Raman amplification allows lower launched powers, which reduces other nonlinear effects such as four-wave mixing
- Raman amplification is broadband and at any wavelength
- Raman amplification can be gain-flattened simply by adjusting the powers of a set of pumps at multiple wavelengths.

Broadband optical noise is generated during Raman amplification by spontaneous Raman scattering. This has a strongly wavelength-dependent noise figure, which is also temperature-dependent.

Rayleigh Scattering

Rayleigh scattering can occur in an optical fiber due to microscopic variations and small inhomogeneities in the refractive index. Light is scattered randomly in all directions, but a small fraction is recaptured in the fiber in the backward direction. The scattered light has the same frequency as an original signal. The backscattered light can also be rescattered to be recaptured in the forward direction. This process is usually called "Double Rayleigh scattering" in the literature. In the case of distributed amplification, the resultant multipass interference (MPI) can produce significant additional noise.

Double Rayleigh scattering is illustrated in Figure 1.

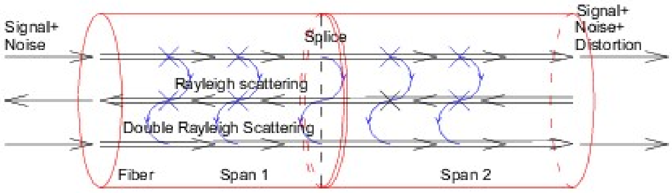


Figure 1 Schematic representation of Double Rayleigh Scattering in optical fibers

Brillouin Scattering

Stimulated Brillouin scattering (SBS) occurs at high intensities. The forward-propagating wave causes a sound-wave by electrostriction. The sound-wave forms a traveling index grating, which reflects the incident lightwave with a downward Doppler shift. SBS is strongest for narrow-band signals, because a strong coherent grating is produced. SBS may be detrimental for optical communication systems, as it strongly reflects unmodulated signals and narrow-band pumps. The strength of SBS depends strongly on the type and bandwidth of modulation of transmitted waves.

The impact of Brillouin scattering on the fiber transmission is illustrated by the demonstrations under *Optical Systems Demos > Characterization > Brillouin Scattering.*

SPM, XPM and FWM

Self-Phase Modulation (SPM) and Cross-Phase Modulation (XPM) originate from the intensity dependence of the refractive index in nonlinear fiber, that is, the temporal variation of the optical intensity of the signal modulates its own phase. SPM is caused by a self-induced phase shift of an optical signal. XPM is due to a copropagating signal at different frequency. SPM and XPM gradually broaden the signal spectrum, so the signal experiences a greater temporal broadening as it propagates along the fiber due to the effects of chromatic dispersion (in frequency region with the normal dispersion).

XPM is usually controlled by increasing the channel spacing or local fiber dispersion. Only adjacent channels contribute significantly to XPM-induced signal distortion in multiple-channel systems.

Four-Wave Mixing (FWM) is due to multiple signals causing variations in refractive index at their difference frequencies. The refractive index then modulates the original carriers to produce sidebands at new frequencies. This interaction can occur between channels in WDM systems, between noise and channels, and between the tones within one channel. FWM is strongest in low-dispersion fibers, because the phases of the carriers and new tones are matched over long lengths.

SPM, XPM and FWM are polarization-dependent. The polarization properties of these effects are included in *UniversalFiber* if Vector Analysis is enabled. For example, XPM causes nonlinear polarization rotation of the Stokes vector of the channels [7].

Polarization Mode Dispersion

Real fibers cannot be perfectly circular and can undergo local stresses (bending and twisting); these random birefringence mechanisms redefine the local birefringence axes along the length, so the propagating light is split into two local polarization modes traveling at different velocities.

The propagation can be described by two principal states of polarization (PSPs): fast and slow. At any point along the fiber, a signal tone will travel in a combination of the fast and slow PSPs. Along the fiber, the signal will scatter between the PSPs, so that signal power in the fast PSP may partially transfer to the slow PSP, for example. The net effect is to cause pulse spreading, which limits the bandwidth of the system.

In standard communication fiber, these asymmetry characteristics (both the orientation of principal birefringence axes and magnitude of birefringence) vary randomly in time, leading to a statistical behavior of PMD. The random change of this birefringence along the fiber length results in random coupling between the modes. This random coupling is represented by random number generators within the model. These can be biased to generate worst cases of PMD.

> **Note:** In *UniversalFiber*, the PMD coefficient is related to the root-mean-squared value of DGD (not to the mean value), that is, $\sqrt{\langle\tau^2\rangle} = D_{PMD}\sqrt{L}$.
> This coefficient is usually provided in fiber specifications and its value is below 0.1 ps/$\sqrt{km}$ for modern fibers. Maximum D_{PMD} coefficients are also usually specified [21].
> This value may be used to simulate the worst cases of linear PMD.

Point Reflections and Losses

Local insertion loss can be caused by reflective and nonreflective events. Reflective events originate from the fact that signal passes through the media with small changes of refractive index. Common sources of reflections are mechanical splices and poorly mated connectors. Additional loss may be caused by mismatches of the numerical apertures and core diameters of connected fibers, or nonconcentric fiber cores, or misalignment of connectors. Loss may be also induced by bends. These are referred to as “event points”. These points are usually represented as discontinuities in the OTDR trace (see Figure 2).

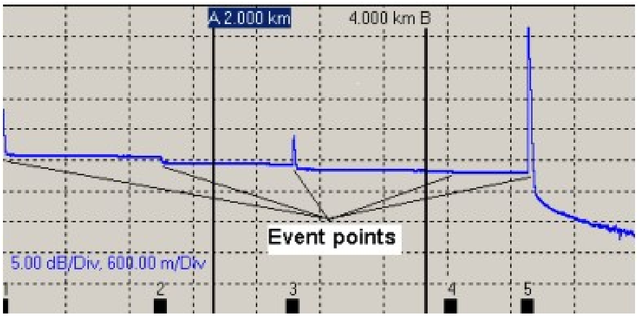


Figure 2 OTDR trace of the fiber link composed of four fiber spans

Analysis Techniques

In this section:

UniversalFiber has two models: Scalar and VectorPMD. The choice between the model is set by the parameter **`PolarizationAnalysis`**.

- **`PolarizationAnalysis`** = `Scalar`: The "scalar model" is a classical model. It assumes that the bands represented by complex envelopes are totally polarized and its polarization remains constant during propagation in the fiber. The band polarization is defined by azimuth and ellipticity. The optical envelopes are represented by only one array which contains sampled values *A(t)*. The equations for the power and field analysis contain only the powers (i.e., S_0 components) of PSs, NBs, discretized bands, and distortions. PMD is not considered. In most practical cases scalar analysis is the best choice, especially when accumulated DGD is small compared with the bit interval. The polarization dependence of nonlinear terms is taken into account only by means of 'adjustment factors'. These factors define how the strength of each nonlinear effect decreases due to random polarization coupling.
- **`PolarizationAnalysis`** = `VectorPMD`: In this case the polarization analysis (that is, polarization-dependent group velocity and polarization properties nonlinear terms) is included into the model. The coarse-step method is applied for field analysis calculations. The fiber model is formed by concatenated sections with constant birefringence. The model is stochastic, so the length and orientation of local axes of birefringence of these sections is defined by two random number generators. The power analysis operates with complete Stokes vectors of the signals. The nonlinear terms are included in the averaged form. This averaging may be done because the length of birefringence sections is chosen to be significantly larger then the fiber correlation length defined by the parameter **`CorrelationLength`**. The bands are represented by two arrays which contain $A_x(t)$ and $A_y(t)$ values.

Model Limitations

Bidirectional signal propagation models are essential in many applications such as discrete Raman amplifiers, systems with distributed Raman gain, WDM systems with bidirectional signal flow, etc., where the linear and nonlinear effects couple signals propagating in opposite directions. For example, the backward Rayleigh scattering results in multiple reflections of the launched signal and amplified noise. Stimulated Raman scattering (SRS) can lead to energy exchange between forward- and backward-propagating waves. Spontaneous Raman scattering of unidirectional signals occurs in both directions, with a frequency shift, and the scattered waves may be further amplified by the SRS.

In such cases, the equations describing signal propagation in the forward direction contain amplitudes of the backward-propagating signals and vice versa. The signal launch configuration defines the boundary conditions for these equations. Usually boundary conditions are specified on both ends of the fiber, at the left end for the forward-propagating wave and at the right end for the backward-propagating wave. A numerical solution of such a two-point boundary problem is difficult because the usual split-step method performs calculations from left to right and thus requires knowledge of the backward-propagating signal on the left fiber end. However, this information is initially unavailable, as mentioned previously.

An iterative method is often used to solve this class of problems. The iterative method, however, is difficult to apply directly for solution of the bidirectional Nonlinear Schrödinger Equations (NLSE) that provide the most accurate description of the signal evolution based on complex amplitudes $A^{\pm}(z,t)$. This is because even one iteration may require a considerable computation time and excessive memory. To resolve this problem a number of reasonable simplifying assumptions are made. The main assumptions and considerations are:

- Dynamic (time and pulse-shape dependent) effects induced by interactions of the counterpropagating waves are neglected. The signals walk through each other with an extremely high velocity. The interaction time of counterpropagating pulses is therefore very short. Thus, the forward-propagating pulse can only respond to the (spatially resolved) average power of the counterpropagating signal $P^{-}(z) = <|A^{-}(z,t)|^{2}>t$. The time shape of the counterpropagating signal is irrelevant.
- Fabry-Perot resonances which build up due to reflections at the opposite fiber ends or due to multiple Rayleigh scattering are ignored. Modeling such effects would require excessively long simulation times significantly exceeding the signal propagation time in fiber. These effects are usually irrelevant in long transmission lines.
- The four wave mixing (FWM) between counterpropagating waves is usually negligible due to the phase mismatch.
- Since the effects of counterpropagating interactions are averaged in time, phase effects such as, for example, cross-phase modulation (XPM), will lead to some constant phase shift (in contrast to XPM of copropagating signals). Such static phase effects are most often of no interest and are neglected.
- Although the power interactions of counterpropagating signals such as, for example, the stimulated or spontaneous Raman scattering or Rayleigh scattering, are also static in time (under the mentioned approximations), they are of high interest in systems design and lead to a change of the signal powers.

Thus, the consideration of interactions between counterpropagating waves will be restricted further to only time-averaged powers which are dependent on the fiber length. To perform such analysis, note that:

- The parameterized signal (PS) representation perfectly fits the needs of such analysis, since it abstracts out the time-dependent and phase issues which are irrelevant in case of counterpropagating waves.
- The PS description is also very effective from the computational point of view. Instead of solving the NLSE in partial derivatives, the power analysis reduces to the ordinary differential equations (ODE) describing the power evolution along the fiber length which can be easily solved using the iterative method described above.
- Power analysis alone is insufficient to describe the dynamic interactions between copropagating waves (for example, FWM, XPM, dispersion) and therefore should be combined with the NLSE description.
- Under the mentioned approximations the NLSE terms providing the coupling of the counterpropagating waves do not directly include the field of the signals propagating in the opposite directions but only require their averaged powers, for example, $P^{-}(z)$. This makes it possible to incorporate them into the NLSE based on the PS representation.

Module Architecture

The following two-stage numerical scheme has been developed:

- In the first stage, a bidirectional Power Analysis is performed, with the two-point boundary problem being solved using the iterative algorithm. For this purpose the complex amplitudes (SFB/MFB signals) at the input fiber ports are replaced by the corresponding parameterized representation.

A schematic representation of the algorithm is shown in Figure 3.

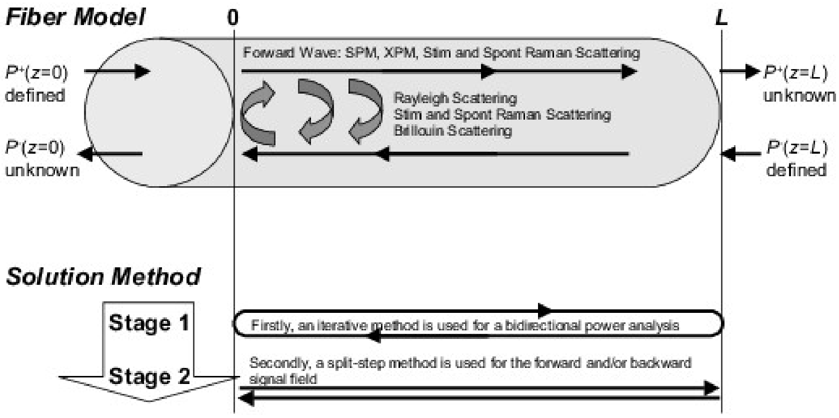


Figure 3 Bidirectional signal propagation in fiber and two-point boundary problem

The simulation speed of this module can be improved using parallel computations on a multicore central processing unit (CPU) or a supported graphical processing unit (GPU). See Chapter 3 in the *VPIphotonics Design Suite™ User Interface Reference* for details on GUI controls for parallel simulations and "GPU-Assisted Simulations" in Chapter 9 of the *VPIphotonics Design Suite™ Simulation Guide* for details on supported hardware.

The *UniversalFiber* module can use parallel computations for the split-step Fourier algorithm. Both FFT calculation and multiplication of the sample arrays will be done in parallel.

When both multithreading within modules and GPU-assisted computation are switched on and supported by the hardware, the module will use the GPU, as it usually reduces the simulation time.

> **Note:** Only the field analysis will be parallelized, both on the CPU and GPU.

Discretizer

The discretizer is the key part of the *UniversalFiber* module that is responsible for taking the optical signals at the inputs of the fiber and converting them to the internal form suitable for using them during the power and field analyses. The following four parameters control the work of the discretizer:

- **`FreqResolutionSFB`** and **`FreqResolutionNB`** define the width of frequency slot for spectrum discretization. In *UniversalFiber* frequency resolution can be specified for bands and NBs separately. This is useful in practical calculations because one can use small value of frequency resolution (for example, less than 100.0e9) for band discretization, but some larger value for noise generation, which prevents the generation of a large number of NBs and prevents a significant increase in simulation time.
- **`GridReferenceFrequency`** is used for alignment of the frequencies of noise bins and sampled band discretization points. The center frequency of one of the generated noise bins is equal to this parameter.
- **`NoiseCenterFrequency`** is the frequency around which NBs will be generated. If there are input NBs, they are transformed to have the same bandwidths as the NBs generated internally.
- **`NoiseBandwidth`** is the width of the frequency region in which NBs are generated by the discretizer. The NBs are generated in the frequency region [**`NoiseCenterFrequency`** − **`NoiseBandwidth`**/2, **`NoiseCenterFrequency`** + **`NoiseBandwidth`**/2]. Input NB outside this range are also treated as noise, but transformed to the same frequency grid as the NBs lying within **`NoiseBandwidth`**.
- **`DiscretizerDescription`** is a switch which defines whether the frequencies of PSs obtained during discretization of input sampled bands are located at the center of frequency interval defined by **`FreqResolutionSFB`** or located at power averaged frequency in **`FreqResolutionSFB`** interval.

The algorithm of discretization depends on the regime in which the module operates. This regime is "controlled" by special enhanced parameters: **`RamanScattering`**, **`RayleighScattering`**, **`BrillounScattering`**, **`SpontaneousRamanScattering`**, **`DiscretizerDescription`**, **`PolarizationAnalysis`**, and **`BidirectionalAnalysis`**.

A schematic representation of spectral discretization is shown in Figure 4.

Rayleigh Distortions

Rayleigh distortions are the signals originating from the Rayleigh backscattering of the signals propagating in the fiber. The word "distortions" is used because, in contrast to parameterized signals, the light represented by Rayleigh distortions does not carry any useful information and leads to degradation of system performance. The frequency of the created distortion is equal to the frequency of the signal for which this distortion has been created. The distortions which are produced due to reflectance at event points are assumed to be Rayleigh distortions too.

Figure 5 shows the spectrum of the forward-propagated band and Rayleigh distortions at the fiber output for both direction.

Brillouin Distortions

WDM signals have the spectrum with pronounced peaks which correspond to the carriers and resonances produced during modulation. The other parts of spectrum have very small energies.

We will consider SBS only for these peaks, as they contain most of the energy. This is justified because SBS takes place at power levels above certain threshold. The value of this threshold can be found analytically when only pump and Stokes wave are interacting, but not with other nonlinear effects like stimulated Raman scattering. Therefore, we are assuming that only the energy of the samples that have the largest power is transformed into the energy of backscattered (Stokes) wave, and the other samples of the band do not take part in SBS.

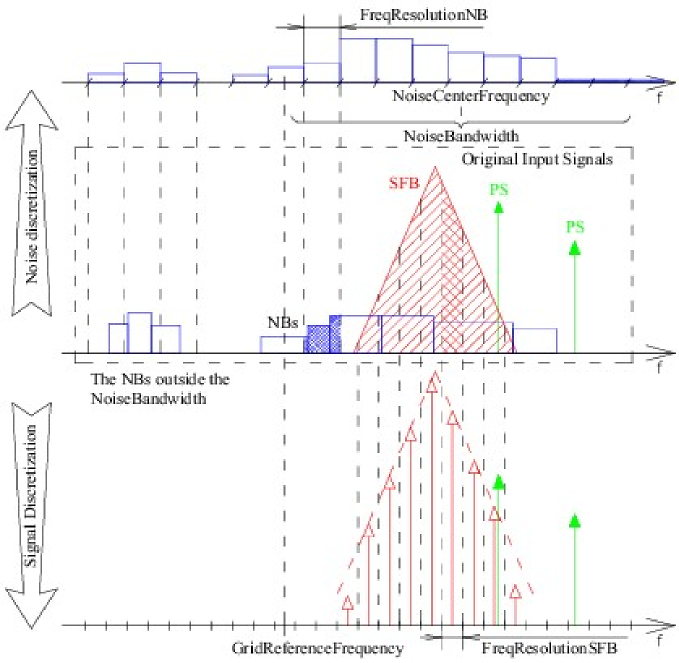


Figure 4 Spectrum discretization for the power analysis

The corresponding high-energy samples will be called *Brillouin PSs*. The counterpropagating Stokes waves will be represented by *Brillouin* distortions:

- Brillouin distortions are treated separately from Rayleigh distortions.
- Brillouin distortions are created for each Brillouin sample and for each input PS.
- Brillouin distortions have frequencies shifted with respect to the frequencies of input bands and PSs.
- Unlike Rayleigh distortions, only one counterpropagating Brillouin distortion is created for each signal.

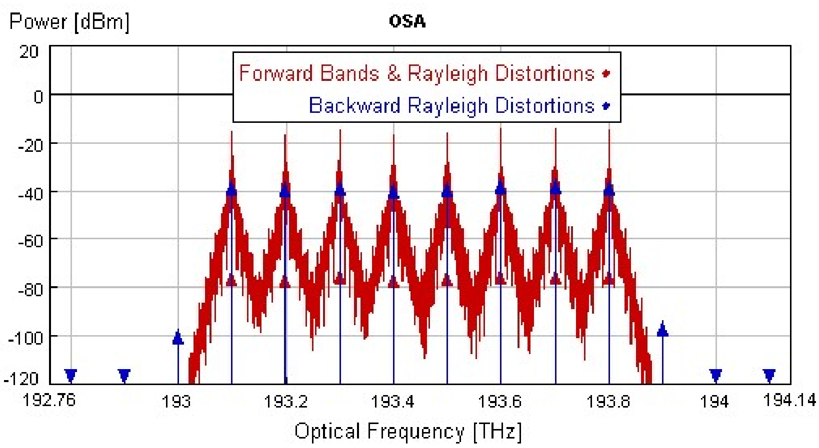


Figure 5 The spectrum of forward-propagated optical signals and Rayleigh distortions created due to Rayleigh backscattering

The parameter **`SBSThresholdFactor`** controls for when SBS should be considered (that is, which waves interact with counterpropagating Brillouin distortions). Figure 6 shows a schematic representation of discretization for a sampled band when SBS is considered. To include more samples that interact with SBS distortions, **`SBSThresholdFactor`** should be increased.

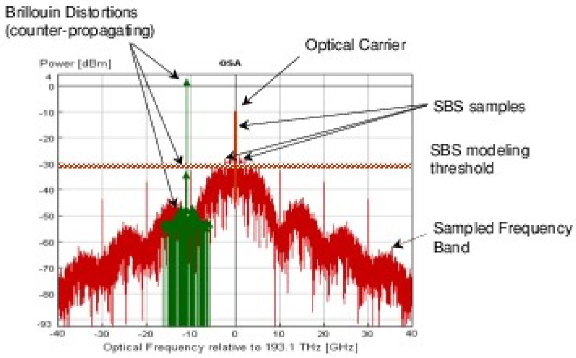


Figure 6 Discretization for sample band when SBS is considered

The model for SBS is semidynamic [24], because the time-domain samples interact with Brillouin distortions, which are parameterized (time-independent).

FWM Distortions

FWM components created when multiple wavelengths mix together are represented by a new type of distortions — *FWM distortions*. The total number of FWM distortions grows dramatically as $N^2(N-1)/2$, where *N* is the total number of interacting signals. FWM distortions are created for both input parameterized signals and parameterized signals created by the discretizer from the input bands. The parameter **FWMThreshold** P_{FWM_Th} is used in order to estimate the strength of FWM. The estimated power of the created distortion given by

$$P = \eta\gamma_{ijkl}^2 P_j P_k P_l exp(-\alpha L) L_{eff}^2 \ , \ \eta = \frac{\alpha^2}{\alpha^2 + \Delta k^2} \cdot \left(1 + \frac{4e^{(-\alpha L)}}{(1 - e^{(-\alpha L)})^2}\right) \ , \ L_{eff} = \frac{1 - exp(-\alpha L)}{\alpha} \qquad (1)$$

is compared with this threshold. If $P < P_{FWM\_Th}$, the distortion is not created and a corresponding term in power analysis is not considered.

- α is a fiber (span) attenuation in (1/m) at the frequency of distortion f_{jkl},
- L is a fiber (span) length,
- γ_{ijkl} is a fiber nonlinear coefficient,
- Δk is a propagation constant mismatch $\Delta k = k_j - k_k + k_l - k_i$. If the propagation constant is expanded to a Taylor series up to third derivatives, Δk can be written as

$$\Delta k = -(2\pi)^2 (f_j - f_k)(f_k - f_l)\left[\beta_2 + 2\pi\beta_3\left(\frac{1}{2}(f_j + f_l - 2f_{ref})\right)\right] \ , \qquad (2)$$

- f_{ref} is defined by the **ReferenceFrequency** parameter.

> **Note:** In the current version of VPI Design Suite, FWM interactions are considered only for the scalar case (PolarizationAnalysis = `Scalar`).

Multiple Fiber Span Parameter Specifications

In order to simulate transmission of optical signals in the fibers composed of multiple spans having different optical characteristics, the parameters of *UniversalFiber* can be specified for each span as arrays. If the number of the fiber spans defined by the parameter **NumberOfFiberSpans** is more than one, these parameters can be specified either as:

- a single value, in which case the same value is applied for every fiber span, or
- as an array of values corresponding to each span in the link.

If any of these parameters are specified by an array, the size of the array must coincide with the number of spans. A similar capability is provided to describe attenuation, Rayleigh backscatter, dispersion and so on using corresponding data files, that is, each of these characteristics can be specified either identically for every fiber span by a single data file or individually for each fiber span using a set of data files.

When multiple file names are provided, they should be separated by spaces. A filename containing spaces can be quoted using double quote marks. For example, two filenames are specified in **RamanFilename**:

```
“C:\Data\Raman Gain SMF.dat” “C:\Data\Raman Gain DCF.dat”
```

For multiple spans, the parameter **FiberLength** defines the length of each fiber span rather than the total fiber length. Note also that the parameter **FiberLength** is ignored if OTDR data is provided (for details refer to Appendix). If OTDR data is provided for many frequencies, the length of the span is calculated as a simple average over all the frequencies.

Scalar Analysis: Powers

Attenuation

Parameter **AttenuationDescription** defines whether an attenuation is specified by a single parameter, in which case the attenuation is assumed to be frequency-independent, or by means of the attenuation file. This parameter may be set to two possible values:

- **AttenuationDescription** = AttenuationParameter: Attenuation is defined by the **Attenuation** parameter [dB/m]. This may be a single value, in which case it is interpreted as attenuation per unit length for every fiber span, or array of value corresponding to each span in the link. The number of specified values should be equal to one or **NumberOfSpans**. If data is imported from OTDR trace file(s), the parameter **Attenuation** is ignored.

  The attenuation coefficient α [1/m] in formulae for field and power analysis is related to the **Attenuation** [dB/m] according to α[1/m] = α[dB/m].ln(10)/10.
- **AttenuationDescription** = AttenuationFile: Frequency-dependent attenuation is read from a file defined by **AttenuationFilename**. The description of the format is in Appendix. A linear interpolation is applied to calculate the attenuations between the specified frequency points. If there are multiple spans, **AttenuationFilename** may have a single filename, which is applied to all spans, or a set of filenames, separated by spaces, corresponding to each fiber span. If data is imported from OTDR trace files for two or more different frequencies, the data from attenuation file is ignored. A more detailed description is given in Import of OTDR Data.

Raman Scattering

Raman scattering is included when **RamanScattering** is set to Yes. The Raman gain profile is read from the file defined by parameter **RamanFilename**. The values responsible for Raman gain are specified in the second column of this file. Files with Raman gain data may contain a single block and a number of blocks. In the first case the file contains the data for one pump frequency. The pump frequency *fp* is specified in the header of the file. The first column is the frequency offset $\Delta f = fp - f \geq 0$, the second column is the Raman gain $g(fp, \Delta f) = g(fp, f) = g(\Delta f)$ for small signal at frequency *f*.

> **Note:** In a single-mode fiber, the Raman gain is usually defined as $g(fp, \Delta f)$ divided by the effective area. This gain relates the powers of the signal and pump, and is therefore experimentally accessible.
>
> The second column in the input file specifies the Raman gain factor $g(\Delta f)$ [1/(W.m)]. This is equivalent to the commonly used Raman Gain Coefficient (RGC, m/W), divided by the effective core area of the fiber (m^2).

Raman gain is defined by the following formula:

$$g(f_k, f_i) = 2\rho\gamma(f_k, f_i)\Im[H(f_k - f_i)] \quad , \qquad (3)$$

where:

- ρ defines Raman contribution to nonlinear response,
- $\gamma(f_k, f_i)$ is a nonlinear coefficient. (For a more detailed description, refer to Vector Analysis: Fields.)
- $\Im(H(f_k - f_i))$ is an imaginary part of normalized Raman response function.

It means that the values $g(f_p, f_p - \Delta f) = 2\rho\gamma(f_p, f_p - \Delta f)\Im[H(\Delta f)]$ are specified in the second column of the Raman file. When two signals at frequencies f_i and f_k interact, the model will rescale the gain to the appropriate pump frequency using the equation

$$g(f_k, f_i) = sgn(f_k - f_i) \cdot \frac{f_i}{f_p - |f_k - f_i|} \cdot \frac{A_{eff}(f_p, f_p - |f_k - f_i|)}{A_{eff}(f_k, f_i)} \cdot g(f_p, f_p - |f_k - f_i|) \quad , \qquad (4)$$

where

- $A_{eff}(f_k, f_i)$ is a reciprocal to overlap integral defined by:

$$\frac{1}{A_{eff}(f_k, f_i)} = f_{ki} = \frac{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} |F_k(x,y)|^2 |F_i(x,y)|^2 dx\,dy}{\left(\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} |F_k(x,y)|^2 dx\,dy\right)\left(\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} |F_i(x,y)|^2 dx\,dy\right)} \quad , \qquad (5)$$

- $F_i(x, y) = F(f_i, x, y)$ is a fiber mode field distribution.

This rescaling is shown in Figure 7.

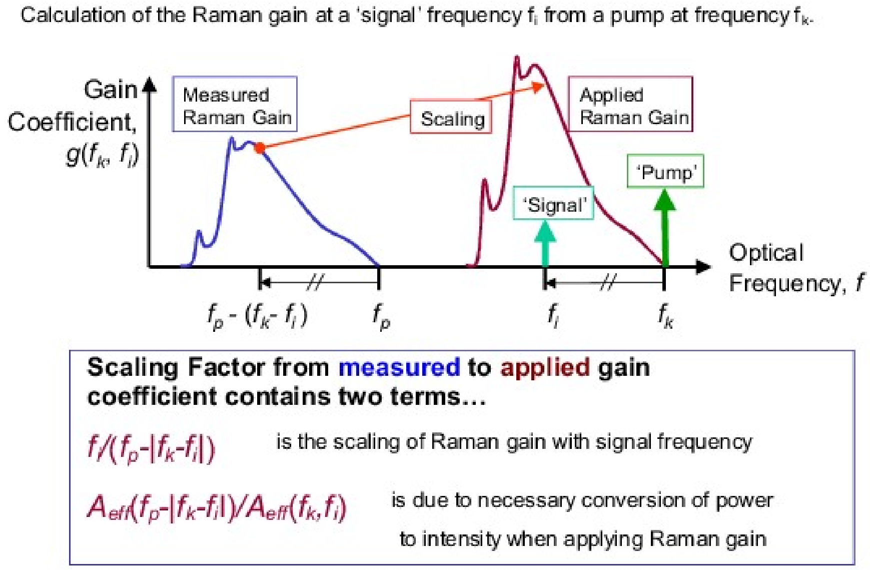


Figure 7 UniversalFiber module rescaling of Raman gain for different pump frequencies

The specification of $A_{eff}(f_k, f_i)$ quantities depends on the switch **`CoreAreaDescription`**. It may be set to three possible values:

- **`CoreAreaDescription`** = `CoreAreaParameter`: in this case, the effective core area is constant for all the frequencies and is defined by the **`CoreArea`** parameter. This switch may be used when the signals are considered to be within a relatively low frequency interval, where the effective core area is nearly constant. This approach may be used for the preliminary tests.
- **`CoreAreaDescription`** = `CoreAreaFile`: the frequency-dependent effective core area is specified by means of the `CoreAreaFile`. The $A_{eff}(f_i, f_i) = A_{eff}(f_i)$ values are written to this file. In order to find $A_{eff}(f_i, f_i)$ we make use of the approximation that the modes are nearly Gaussian, that is,

$$\tilde{A}_{eff}(f_k, f_i) = \frac{1}{2}(A_{eff}(f_k) + A_{eff}(f_i)) \quad . \tag{6}$$

- **`CoreAreaDescription`** = `OverlapIntegralFile`: frequency-dependent overlap integrals $f_{ki} = f(f_k, f_i)$ are specified in the file defined by the parameter **`OverlapIntegralFilename`**. Bilinear interpolation [29] is used to find the overlap integral among two-dimensional data sets.

> **Note:** The manufacturers very often provide Mode Field Diameter (MDF) 2w fiber parameter that represents a measure of the transverse extent of the electromagnetic field intensity of the mode in a fiber cross section [12]. This parameter may be measured in the experiment (for example, using the variable aperture test method). Using the Gaussian approximation for fiber modes, the effective area may be found using $A_{eff} = \pi w^2$ formula. A more general but empirical relationship between A_{eff} and w is $A_{eff} = \kappa\pi w^2$, where coefficient κ is the correction factor which depends on the wavelength. This approximation is accurate for G.652 and G.654 step-index fibers near the LP_{11} cut-off, but for G.652 and G.654 fibers at much longer wavelengths and in the case of G.653 dispersion shifted fibers, A_{eff} cannot be accurately estimated. For example, for SMF−28 fiber 2w = 9.2 ± 0.4 at 1310 nm (A_{eff} = 66.5 μm^2) and 10.4 ± 0.8 at 1550 nm (A_{eff} = 84.9 μm^2).

> **Note:** The Raman gain factor is highly dependent on the Ge-doping of the fiber and its core area, and is large for high-Ge small-core fibers such as Dispersion Compensating Fibers (DCF).

The Raman file may consist of a number of blocks (two or more). Each block represents a single Raman gain curve measured at some definite pump frequency. Each block starts with a block separator, specifying frequency or wavelength of the pump wave, followed by measured data. To calculate the Raman gain produced by a pump wave with frequency f_k affecting a signal wave with frequency f_i, the module applies a bilinear interpolation.

> **Note:** The files with single blocks and the ones with multiple blocks have similar file formats — the only difference

is the number of blocks.

Figure 8 shows Raman gain spectra for different fiber types measured by NIST [15] for 1460 nm pump.

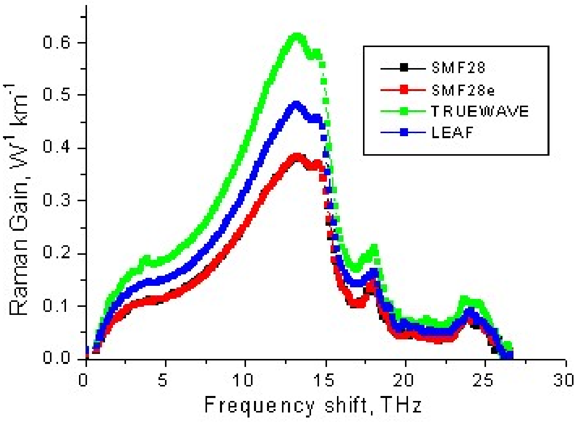


Figure 8 Raman gain coefficient for SMF28, SMF28e, TRUEWAVE, and LEAF fibers measured for 1460 nm pump

Raman gain depends strongly on the state of polarization of the interactive waves. Mainly only copolarized components are amplified. In typical fibers the state of the polarization does not preserve during the propagation, so averaged gain coefficient is approximately a half of that measured over a short distance. *UniversalFiber* includes adjustment factors defined by the parameter **`RamanAdjustmentFactors`**. The factor defines how the Raman gain reduces because of random polarization scrambling.

To avoid strong polarization dependence of Raman gain, the pump and probe are usually depolarized in the experiments. For gain coefficients measured in this way, the polarization correction factors should be set to 1.0. If Raman gain is measured in a short distance, where probe signal and pump are colinear polarized, these factors should be set to 0.5. In the general case, these factors may be used to fit experimental data to computation results, or to switch on/off individual terms.

Adjustment factors for stimulated (ST) and spontaneous (SP) Raman scattering can be specified for interactions between signals (S), noise (N), and distortions (D) (refer to Table 2 for details).

Table 2 ***The nine adjustment factors entered in RamanAdjustmentFactors***

Raman	Stimulated			Spontaneous
.../...	Signal	Noise	Distortion	Noise
Signal	ε_0	ε_1	ε_2	ε_6
Noise	ε_1	ε_3	ε_4	ε_7
Distortion	ε_2	ε_4	ε_5	ε_8

`RamanAdjustmentFactors` may be specified in four possible ways:

- If one factor is specified, it is used to scale the interaction between all kinds of signals. It means that all the ε_i are equal.
- If two factors are specified, the first one corresponds to the stimulated, and the second, to spontaneous Raman interactions, $\varepsilon_0 = \varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon_4 = \varepsilon_5$, $\varepsilon_6 = \varepsilon_7 = \varepsilon_8$.
- If five factors are entered, they correspond to 1) S–S ST 2) S–N, S–D ST 3) N–N, N–D,D–D ST 4) S–N SP 5) N–N, D–N SP interactions, $\varepsilon_1 = \varepsilon_2$, $\varepsilon_3 = \varepsilon_4 = \varepsilon_5$, $\varepsilon_7 = \varepsilon_8$.
- If nine factors are entered, they correspond to 1) S–S ST 2) S–N ST 3) S–D ST 4) N–N ST 5) N–D ST 6) D–D ST 7) S–N SP 8) N–N SP 9) D–N SP interactions.

Note: The parameter **`RamanAdjustmentFactors`** is used only when **`PolarizationAnalysis`** is set to `Scalar`.

Noise bins corresponding to spontaneous Raman scattering will be generated, if the parameter **`SpontaneousRamanScattering`** is set to `Yes`.

Parameter **`NoiseBinInteractions`** defines whether the Raman power transfer from noise bins to signals, distortions, and other noise bins is neglected (set to `No`) or considered (`Yes`). **`DistortionInteractions`** has a similar meaning, but it switches on/off power transfer from distortions to signals, noise bins and other distortions. These approximations are usually acceptable when the signal power is much greater than the noise or distortion power. When these parameters are set to `No`, the corresponding terms are excluded from equations for power and field analysis, which can speed up the computation significantly because of the large number of noise bins and distortions.

> **Note:** Evidently, when this feature is activated, the equations are solved explicitly, but it is acceptable for most simulations in practise. Use these two switches for preliminary calculations.

For a detailed description of the format of Raman files, see Appendix.

Brillouin Scattering

Brillouin scattering is considered if the **`BrillouinScattering`** parameter is set to `Yes`. Each PS *(input PS or Brillouin PS)* and each distortion *(Brillouin, Rayleigh, FWM or input distortion)* at frequency f_i interacts with all opposite-propagated PSs and distortions lying within the $[f_i - f_B - \Delta f_B,\ f_i - f_B + \Delta f_B]$ and $[f_i + f_B - \Delta f_B,\ f_i + f_B + \Delta f_B]$ intervals.

Brillouin gain is assumed to have a Lorentzian spectral profile given by:

$$G_B(f_k, f_i) = C_B \frac{g_B(f_i)}{A_{eff}(f_i, f_k)} \frac{(\Delta f_B/2)^2}{(f_k - f_i - f_B)^2 + (\Delta f_B/2)^2}, \qquad (7)$$

where:

- $g_B(f_i)$ is a SBS gain at frequency f_i.
  In contrast to the specification for Raman gain, SBS gain is specified relative to the intensities of interacting waves and "actual" modal gain is defined by $g(f_i)/A_{eff}(f_k, f_i)$. In the literature, gain is usually specified in [m/W] units.
- C_B is SBS correction factor used only for PS (its value varies in (0, 1] interval). This parameter defines the gain suppression value due to the modulation.
  SBS gain depends on the modulation type of the signal and the ratio $T*\Delta fB$, where ΔfB is a SBS bandwidth, $1/T$ is a bit rate.

It is well known that by modulating the signal carrier, stimulated Brillouin scattering is reduced. High modulation rates produce broad optical spectra and this reduction can be expected. The results depend on the particular encoding scheme used: ASK, FSK or PSK, and on the ratio $B/\Delta fB$. The information about the modulation format is specified in the label data. The SBS correction factor is read from the property **`SBSCorrectionFactor`** of the PS. This property may be set using the *SetBrillouinCorrectionFactor* module in the *Signal Conversion* folder.

- $A_{eff}(f_i, f_k)$ is a reciprocal of the overlap integral defined by formula (5),
- Δf_B is a spectral width of Brillouin gain profile equal to full width at half maximum. It is defined by parameter **`SBSBandwidth`**.
- f_B is a Brillouin frequency shift defined by parameter **`SBSStokesShift`**. The value will be quantized to 1/TimeWindow frequency grid.
- ε_B is a SBS polarization adjustment factor defined by parameter **`SBSAdjustmentFactor`**. This factor defines how SBS gain reduces due to random polarization scrambling.

> **Note:** The parameter **`SBSAdjustmentFactor`** is used only when **`PolarizationAnalysis`** is set to `Scalar`.

There are two opportunities to specify SBS gain. Parameter **`SBSDescription`** defines how the Brillouin gain profile is defined and may be set to:

- **`SBSDescription`** = `SBSGainParameter`, in which case the Brillouin gain is assumed to be frequency-independent and defined by the **`SBSGain`** parameter; or
- **`SBSDescription`** = `SBSGainFile`, in which case the frequency-dependent Brillouin gain is read from a file defined by the parameter **`SBSGainFilename`**.

Both the file and parameter may be a single value, in which case they are referred for every fiber span, or an array of values corresponding to each span in the link.

FWM

The terms describing FWM interactions in power analysis equations are given by

$$\frac{dP_i}{dz} = \begin{pmatrix} 4(1-\rho)\xi_6 \sum\limits_{\substack{f_i = f_j - f_k + f_l \\ j, l \neq k}} \frac{\gamma_{ijkl}}{(1+\delta_{jl})}\sqrt{P_iP_jP_kP_l}\sin(\phi_{ijkl}) + \\ +\xi_7 \sum\limits_{\substack{f_i = f_j - f_k + f_l \\ j, l \neq k}} \frac{\sqrt{P_iP_jP_kP_l}}{(1+\delta_{jl})}[\Im(G_{ijkl}(\omega_k,\omega_j) + G_{ijkl}(\omega_k,\omega_l))\cos(\phi_{ijkl}) + \\ +2\Re(G_{ijkl}(\omega_k,\omega_j) + G_{ijkl}(\omega_k,\omega_l))\sin(\phi_{ijkl})] \end{pmatrix} \quad (8)$$

(the second term is used when the parameter **RamanScattering** is set to Yes).

The equation (8) uses the following values:

- P_i, P_j, P_k, P_l are the signals and FWM distortions propagated either in the forward or backward directions,
- ξ_i are the polarization adjustment factors defined by the **NonlinearAdjustmentFactors** parameter,
- $\phi_{ijkl} = \phi_{ijkl}^{(L)} + \phi_{ijkl}^{(NL)}$ is a phase mismatch consisting of two parts – linear and nonlinear.

$$\phi_{ijkl}^{(L)} = -\Delta k_{ijkl} z \ , \quad (9)$$

Δk_{ijkl} is defined by

$$\Delta k = -(2\pi)^2 (f_j - f_k)(f_k - f_l)\left[\beta_2 + 2\pi\beta_3\left(\frac{1}{2}(f_j + f_l - 2f_{ref})\right)\right] \ . \quad (10)$$

- $\phi_{ijkl}^{(NL)} = \phi_j^{(NL)} - \phi_k^{(NL)} + \phi_l^{(NL)} - \phi_i^{(NL)}$ is a nonlinear phase mismatch between i, j, k, and l interacting signals. The nonlinear phase shift of each signal is determined by simulated nonlinear effects:

$$\frac{d\phi^{(i)}}{dz} = \sum_{NL\ effects} \frac{d\phi_{NL\ effect}^{(i)}}{dz} \ ,$$

where particular nonlinear effects can be switched on or off using the module parameters, **SPM_EC**, **XPM_EC**, **XPM_MC**, **IntraBandRaman**, **InterBandRaman**.
When any of these parameters is set to Yes, the respective equation (11)–(15) is used:

$$\frac{d\phi^{(i)}}{dz}_{SPM} = -(1-\rho)\xi_1\gamma_{ii}P_i \ , \quad (11)$$

$$\frac{d\phi^{(i)}}{dz}_{IntraBandRaman} = -\rho\xi_2\gamma_{ii}P_i \ , \quad (12)$$

$$\frac{d\phi^{(i)}}{dz}_{XPM\_EC} = -2(1-\rho)\xi_3\sum_{k\neq i}\gamma_{ki}P_k \ , \quad (13)$$

$$\frac{d\phi^{(i)}}{dz}_{XPM\_EC+XPM\_MC} = -\rho\xi_4\sum_{k\neq i}\gamma_{ki}P_k \ , \quad (14)$$

$$\frac{d\phi^{(i)}}{dz}_{InterBandRaman} = -\frac{\xi_5}{2}\sum_{k\neq i}\Re(G_{ki}(\omega_k-\omega_i))P_k \ , \quad (15)$$

- Two more terms are the phase counterparts of equation (8) and define the nonlinear phase shift induced by the four-wave mixing:

$$\frac{d\phi^{(i)}}{dz}_{FWM\_Kerr} = -2(1-\rho)\xi_6 \sum_{\substack{j,k,l \\ \omega_i = \omega_j - \omega_k + \omega_l}} \frac{\gamma_{ijkl}}{(1+\delta_{jl})}\frac{\sqrt{P_iP_jP_kP_l}}{P_i}\cos(\Delta\phi_{ijkl}) \ , \quad (16)$$

$$\frac{d\phi^{(i)}{}_{FWM\_Raman}}{dz} = \frac{\xi_7}{2} \sum_{\substack{j,k,l \\ \omega_i = \omega_j - \omega_k + \omega_l}} \frac{1}{(1+\delta_{jl})} \frac{\sqrt{P_j P_k P_l}}{\sqrt{P_i}} \begin{pmatrix} (\Im(G_{ijkl}(\omega_k, \omega_j)) + \Im(G_{ijkl}(\omega_k, \omega_l)))\sin(\Delta\phi_{ijkl}) \\ -2(\Re(G_{ijkl}(\omega_k, \omega_j)) + \Re(G_{ijkl}(\omega_k, \omega_l)))\cos(\Delta\phi_{ijkl}) \end{pmatrix} \quad (17)$$

Equation (17) is used when **RamanScattering** = Yes.

- $G_{ijkl}(\omega_k, \omega_j)$ is a Raman gain

$$G_{ijkl}(\omega_k, \omega_j) = G_{kj}(\omega_k, \omega_j) \cdot \frac{A_{kj}}{A_{ijkl}} , \quad (18)$$

in which A_{ijkl} is given by (20).

- γ_{ijkl} is a nonlinear coefficient

$$\gamma_{ijkl} = \gamma_{ii} \cdot \frac{A_{ii}}{A_{ijkl}} . \quad (19)$$

- A_{ijkl} is an overlap integral for four interacting signals

$$A_{ijkl} = \frac{1}{4}\sqrt{A_{ii}A_{jj}A_{kk}A_{ll}}\left(\frac{1}{A_{ii}} + \frac{1}{A_{jj}} + \frac{1}{A_{kk}} + \frac{1}{A_{ll}}\right) . \quad (20)$$

In contrast to other effects simulated during the power analysis, the four-wave mixing is dependent on the phase matching between interacting signals. The *UniversalFiber* module assigns zero phase to all input signals and then simulates the phase changes according to the equations above. However, the parameterized signals (and distortions) that are passed between different modules contain no phase information.

This leads to the following limitations of FWM simulation:

- When several *UniversalFiber* modules are connected to each other, the phase of the parameterized signals and distortions is lost when signals are transferred between the fiber modules. Typically, this leads to *overestimation* of power of FWM products in multiple fiber links, because the input signals are treated as *in-phase* at the module input and the phase-matching conditions are better. At the same time, depending on the length of the fiber, signal powers, etc., multiple fiber link simulation can lead to *underestimation* of FWM power.

  This problem does not occur in a single fiber module with no input FWM distortions, even a multiple-span fiber is simulated properly.
- Because of the phase-sensitive nature of FWM, when some parameterized signals have *the same* frequency, an ambiguity appears: it is possible to treat the signals either coherently or incoherently, producing different powers of FWM products.

  To overcome this issue, the *UniversalFiber* module uses the following logic:
  + If there are *input* parameterized signals with the same frequency, the module issues an *error*. To continue with the simulation, it is necessary to join (or convert) the signals using the *SignalConverter* module. Alternatively, part of the input signals can be converted to crosstalk or generic distortions, depending on the simulation.
  + An input parameterized signal can sometimes coincide in frequency with the signal resulting from the input sampled band discretization (see Discretizer for details). The signal overlap can be eliminated in principle using the spectral discretization parameters of the module.

    Using **SpectralDiscretizerDescription** = AverageWeighted normally helps to avoid spectral overlap between input and discretized signals, but usually causes a large number of FWM products to be simulated (because of nonuniform frequency spacing).

    Normally, the input and discretized signals will have quite different power, and influence of overlapping signals on resulting FWM products is negligible. For 'internal' signal overlap, the module issues a warning and then uses a modified version of the algorithm above to exclude nonphysical FWM products.

Rayleigh Scattering

Rayleigh scattering is included into consideration if parameter **RayleighScattering** is set to Yes.

Parameter **BackscatterDescription** specifies whether the backscatter is entered as the proportion of power reflected per meter, or as the measured reflection of a pulse with defined width. This switch defines the type for specified units for both Rayleigh parameter and data from the Rayleigh file. It may be set:

- **BackscatterDescription** = OTDR: In this case Rayleigh coefficient is measured in dB units, usually provided by fiber manufactures. Physical sense of backscatter coefficient measured in dB is the part of impulse reflected back after passing the distance, equal to the pulse width. The linear Rayleigh coefficient used in formulae for power analysis is:

(21)

$$\eta[1/m] = \frac{n_{eff}}{c\tau} 10^{\frac{\eta[dB]}{10}}$$

where:

+ n_{eff} is an effective group index defined by module parameter **EffectiveGroupIndex**,
+ c is the speed of light,
+ τ is a pulse width for OTDR measurements, defined by parameter **OTDRPulseWidth**. It is usually assumed to be 1 um or 1 ns.

- **BackscatterDescription** = Linear: Rayleigh coefficient is measured in linear units and defined as a proportion of power reflected per specified length.

Parameter **RayleighDescription** defines how the fiber Rayleigh backscattering coefficient is specified: by single frequency-independent parameter or by the file and can be set to:

- **RayleighDescription** = RayleighCoefficient: Rayleigh coefficient is defined by module parameter **RayleighBackscatterCoefficient**. If **BackscatterDescription** = Linear, this parameter specifies the backscattering coefficient as the proportion of power reflected per meter (1/m units). If **BackscatterDescription** = OTDR, this parameter specifies the backscattering coefficient as the measured reflection of a pulse with defined width (dB units). This may be a single value, in which case it is interpreted as the backscatter coefficient of every fiber span, or an array of values corresponding to each span in the link. If data is imported from OTDR trace file(s), this parameter is ignored.
- **RayleighDescription** = RayleighFile: Rayleigh coefficient is specified in file defined by module parameter **RayleighFilename**. If there are multiple spans, this may be a single filename, which is applied to all spans, or a set of filenames, separated by spaces, corresponding to each fiber span. Parameter **BackscatterDescription** defines which sort of data is written to the file: if it is set to Linear, frequency-dependent Rayleigh coefficients are specified in relative units per specified length (1/m, or 1/km, or dB/m, or dB/km), if it is set to OTDR, in dB units. In later case, formula (21) should be used. If the data is imported from OTDR trace files for two or more different frequencies, the data from specified Rayleigh file is ignored. A more detailed description is given in Import of OTDR Data.

Loss and Reflection Events

The fiber may be composed of multiple spans, a number of distinct sections being specified by the parameter **NumberOfFiberSpans**. Practically all physical parameters may be specified for each span separately. At the boundaries of fiber spans (or event points), signals attenuate and reflect back. These effects are included by means of parameters that describe local loss and reflectance.

Parameter **EventLossDescription** defines how the fiber event loss is specified. It can be set to:

- **EventLossDescription** = EventLossParameter: in this case frequency-independent event loss value(s) **EventLoss** is used. If the information is read from the OTDR file(s), parameter **EventLoss** is ignored (except the values for the first and the last events).
- **EventLossDescription** = EventLossFile, in which case the frequency-dependent event loss is read from a file defined by parameter **EventLossFilename**. Note that this file may contain information not only about the event insertion loss a_m[dB](f_i, L_m), but also about transmission coefficient T (if specified in the file header):

$$T(f_i, L_m) = 10^{-\frac{a_m[dB](f_i, L_m)}{10}} \quad . \qquad (22)$$

In both cases they may be single values equal for each event point, or an array of values corresponding to each event point of the link. If an array is provided, then the number of values must be equal to the number of fiber spans plus one.

Parameter **EventReflectanceDescription** defines how the fiber event reflectance is specified. This can be set to:

- **EventReflectanceDescription** = EventReflectanceParameter: in this case frequency-independent **EventReflectance** value is used. If the information is read from the OTDR file(s), parameter **EventReflectance** is ignored (except the values for the first and the last events).
- **EventReflectanceDescription** = EventReflectanceFile, in which case the frequency-dependent event reflectance coefficient is read from a file defined by parameter **EventReflectanceFilename**. By analogy with event loss, reflectance dependence on frequency (wavelength) may be specified in the data file either as a reflectance in dB or a dimensionless reflection coefficient (see (22) and Table).

The information of event loss and reflectance at the first and last event points is usually used in calculations and never ignored. Alternatively, event reflectance characteristics can be imported from OTDR data file(s) directly. A more detailed description is given in Import of OTDR Data.

> **Note:** The parameters describing event loss do not take into account return loss caused by event reflection. We assume that event loss is referred to nonreflective events, event reflectance, to reflective.

Polarization Issues

In many cases it would be important and sufficient to include the polarization dependence of the power effects in Raman and Brillouin interactions. For this purpose the module can be executed in two polarization modes:

- The first mode (**NoisePolarization** = `X` or `Y`) assumes that excess noise generated via spontaneous Raman or Brillouin scattering is simulated in a single polarization. That is, half the noise power is left out of the simulation. (Refer to *TestSetAmplifier* for a detailed discussion.) Since the SRS for orthogonally polarized waves is normally considered to be negligible, the orthogonal polarization component does not change the power balance and can be ignored in the power analysis. Note, however, that because the orthogonal polarization component is neglected, the total amount of noise power at the fiber output is less than it would be in reality. The module behaves as if it has a polarizer placed at the output. Please take into account that this may affect OSNR calculations, where it is important to distinguish between cases with or without polarization filtering.

  If the polarization of the input noise bins is different from the x or y polarizations, then it is converted to x or y polarization, and a warning message is shown.
- In the second mode (**NoisePolarization** = `Unpolarized`) it is assumed that the *noise* due to spontaneous Raman scattering is unpolarized, that is, it has a zero degree of polarization. Generated noise is simulated in both polarizations since both orthogonal polarization components are generated, the total rate of the excess noise power is doubled in comparison to the linear polarization case. However, the Raman gain affects only one polarization component of the noise at any point along the fiber; therefore, the effective Raman gain should be only a half of its value for the equal polarizations. This correction of the Raman gain for noise waves can be done manually by setting the second adjustment coefficient ε_2 of parameters. It should be noted that due to the

  preferable Raman amplification of the polarization component which coincides with the SOP of the signal waves, the assumption of the unpolarized noise becomes invalid during the signal propagation. However, due to the polarization mode dispersion that occurs in practice, the SOP of both noise and signals rapidly becomes uncorrelated and quickly varies over the fiber length. This shows that the above assumption may be reasonable in the statistical sense, when considering long transmission distances.
- If the **DistortionPolarization** = `Unpolarized`, the generated distortions will have zero degree of polarization. If the input distortions are not unpolarized, their degree of polarization will be set to zero.
- **DistortionPolarization** = `X` or `Y`, the distortions will have x or y output polarization.

Scalar Analysis: Fields

The field analysis is based on the frequency decomposition approach. It has been extended for bidirectional propagation. Complex amplitudes are used in the field equations for all copropagating signals arriving at the fiber inputs in form of SFB/MFB. The parameterized description is used for copropagating PSs, NBs and all counterpropagating signals, in particular those originally represented by SFB/MFB. For clarity the field equations below are divided into the linear part, including the frequency-dependent attenuation and dispersion, and the nonlinear part which, due to the variety of interaction types and a large number of terms, is described separately.

The parameter **FieldAnalysis** is used to specify if the field analysis must be done

- for both forward- and backward-propagating bands (**FieldAnalysis** = `Bidirectional`),
- only for forward-propagating bands (**FieldAnalysis** = `Forward`), which is useful for speeding up the calculations when the exact result for the backward-propagating signals is not important,
- only for backward-propagating bands (**FieldAnalysis** = `Backward`), or
- the field analysis is not needed (**FieldAnalysis** = `No`).

If field analysis in some direction is not performed, the band is filtered using the filter function constructed using input and output powers of PS created during the band discretization.

The parameter **ConserveMemory** specifies if the algorithm should store the intermediate results in memory during the field analysis. Setting this parameter to `No` makes the module work faster if sufficient memory is available. However, in cases when the number of frequency bands is large and each band contains a significant number of samples, **ConserveMemory** should be set to `Yes` in order to prevent running out of memory.

The generalized nonlinear Schrödinger equation describing a propagation of the *i*-th sampled frequency band (SFB) with center frequency f_i and propagating in the forward direction is given by:

$$\frac{\partial}{\partial z}A_i^+(t) = \underbrace{-\frac{\alpha}{2}A_i^+(t)}_{attenuation} + \underbrace{\ddot{D}A_i^+(t)}_{dispersion} + \underbrace{\ddot{N}A_i^+(t)}_{nonlinearity}, \tag{23}$$

where:

- $A_i^+(t)$ is a complex envelope for i channel (input signal) with center frequency f_i,
- t is referred to retarded time,
- α is a frequency-dependent attenuation coefficient,
- D is a differential operator responsible for dispersion in a linear medium,
- N is a nonlinear operator that governs the effects of all the fiber nonlinearities on pulse propagation.

Linear Operator

A linear operator is often considered in frequency domain and includes attenuation and dispersion.

Attenuation

Attenuation is included in linear operators by means of the power attenuation parameter. For more details, refer to the power analysis section.

Dispersion

The dispersion operator can be characterized either by specifying the group velocity **Dispersion** (GVD) coefficient and **DispersionSlope** at the **ReferenceFrequency** (**DispersionDescription** = `DispersionParameters`), or by loading data for dependence of GVD vs. wavelength from the file defined by parameter **DispersionFilename** (**DispersionDescription** = `DispersionFile`).

- **DispersionDescription** = `DispersionParameters`: the dispersion operator is defined as:

$$\hat{\underline{D}} = j\frac{\beta_2}{2}\frac{\partial^2}{\partial t^2} + \frac{\beta_3}{6}\frac{\partial^3}{\partial t^3}\ , \tag{24}$$

  which in frequency domain is transformed to:

$$\hat{\underline{D}} = j\left[\frac{\beta_2(\omega_{ref})}{2}(\omega - \omega_{ref})^2 + \frac{\beta_3(\omega_{ref})}{6}(\omega - \omega_{ref})^3\right]\ , \tag{25}$$

  where

$$\beta_1(\omega) = \frac{d}{d\omega}\beta(\omega)\ ,\ \beta_2(\omega) = \frac{d^2}{d\omega^2}\beta(\omega)\ ,\ \beta_3(\omega) = \frac{d^3}{d\omega^3}\beta(\omega)\ , \tag{26}$$

$$\beta_2 = -\frac{c}{2\pi f_{ref}^2}D_\lambda\ ,\ \beta_3 = \frac{c}{(2\pi)^2 f_{ref}^3}\left(\frac{c}{f_{ref}}S_\lambda + 2D_\lambda\right)\ , \tag{27}$$

  D_λ is a fiber dispersion defined by parameter **Dispersion**,

  S_λ is a dispersion slope defined by parameter **DispersionSlope**.

  Note: Both these values are specified at the **ReferenceFrequency**.

- **DispersionDescription** = `DispersionFile`: in this case the dependence of dispersion at optical frequency is loaded from the file specified by **DispersionFilename**. In the frequency domain, the dispersion operator is given by:

$$\hat{D}(\omega) = -j\,[\beta(\omega) - \beta(\omega_{ref}) - \beta_1(\omega_{ref})(\omega - \omega_{ref})]\ . \tag{28}$$

The dispersion operator (28) is defined so that the phase shift and group delay vanish for an optical signal whose carrier frequency is equal to the fiber **ReferenceFrequency**.

Taking into account (26), (27), (28) we obtain

$$\hat{\underline{D}}(\omega) = \int_{\omega_{ref}}^{\omega} d\omega' \int_{\omega_{ref}}^{\omega'} \beta_2(\omega'')\,d\omega''\ , \tag{29}$$

where

The equations are solved in the coordinate system that moves with the propagating optical pulse.

Typical values of dispersion profiles are shown in Figure 9.

The accumulated value of dispersion is also stored to the corresponding "Dispersion" label of PS and may be later visualized by dispersion probe modules.

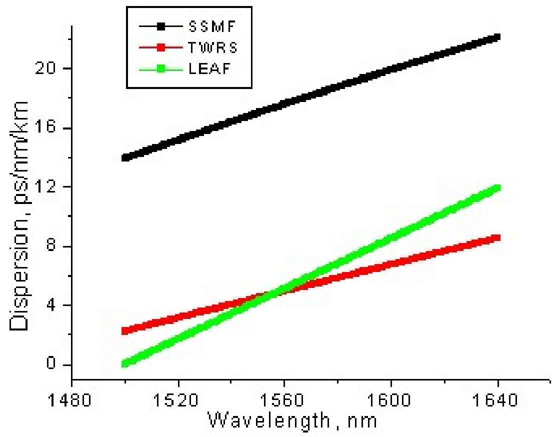


Figure 9 Dispersion profiles for different fiber types

Nonlinear Operator

The nonlinear part of field analysis equations for forward-propagating bands is

$$\hat{N}A_i^+(t) = -j\Big[\underbrace{\xi_1(1-\rho)\gamma_{ii}\left|A_i^+(t)\right|^2A_i^+(t)}_{SPM\_EC} + \underbrace{\xi_2\rho\gamma_{ii}A_i^+(t)\int_0^\infty h(\tau)\left|A_i^+(t-\tau)\right|^2d\tau}_{IntraBandRaman} + \tag{31}$$

$$\underbrace{\xi_3 2(1-\rho)A_i^+(t)\sum_{k\neq i}\gamma_{ki}\left|A_k^+(t)\right|^2}_{XPM\_EC} + \underbrace{\xi_4\rho A_i^+(t)\sum_{k\neq i}\gamma_{ki}\int_0^\infty h(\tau)\left|A_k^+(t-\tau)\right|^2d\tau}_{XPM\_MC} + \tag{32}$$

$$\underbrace{\xi_5\rho A_k^+(t)\sum_{k\neq i}\gamma_{ki}\int_0^\infty h(\tau)A_i^+(t-\tau)A_k^{+*}(t-\tau)e^{2\pi j(f_i-f_k)\tau}d\tau}_{InterBandRaman} + \tag{33}$$

$$+\underbrace{\rho\sum_{k\neq i}\gamma_{ki}\int_0^\infty h(\tau)A_i^+(t-\tau)[\varepsilon_0P_S^\pm(f_k)+\varepsilon_2P_{RD}^\pm(f_k)+\varepsilon_2P_{BD}^\pm(f_k)+\varepsilon_1P_N^\pm(f_k)]e^{2\pi j(f_i-f_k)\tau}d\tau}_{InterBandRaman}\Big] \tag{34}$$

where:

- $A_i^+(t)$ is an amplitude of the *i*-th forward-propagating sampled band, $A_k^+(t)$, *k*-th forward-propagating sampled band.
- $\gamma_{ik} = \gamma(f_k, f_i)$is a nonlinear coefficient that defines interactions of *i*-th and *k*-th propagating bands,
- ρ is the coefficient which defines the fractional contribution of the delayed Raman response function governed by $h(\tau)$. Raman response function can be obtained from the Raman gain spectrum which can be measured experimentally.
- $h(\tau)$ is a normalized function responsible for Raman nonlinear contribution, $\int_0^\infty h(\tau)d\tau = 1$.

Nonlinear effects are switched on/off by special switches defined by enhanced parameters: `SPM_EC`, `XPM_EC`, `XPM_MC`, `IntraBandRaman`, `InterBandRaman`.

The first terms in (31) and (32) are responsible for instantaneous electronic nonlinear contribution (Kerr nonlinearity). This effect is characterized by the nonlinear coefficient γ_{ki} defined by:

$$\gamma_{ki} = \gamma(f_k, f_i) = \frac{2\pi n_2 f_i}{cA_{eff}(f_k, f_i)} \ , \tag{35}$$

where:

- n_2 is a coefficient that defines the value of nonlinear addition to the refractive index,
- c is the speed of light,
- $A_{eff}(f_k, f_i)$ is a reciprocal to overlap integral defined by formula (5):
- $A_{eff}(f_i, f_i)$ is an effective core area of the fiber at frequency f_i usually specified by manufactures.

> **Note:** The value of the nonlinear index n_2 used in this module assumes that it is measured with a constant linear polarization throughout the measured fiber. For fibers with randomly varying birefringence and **`PolarizationAnalysis`** = `Scalar` the nonlinear terms should be corrected using the parameter **`NonlinearAdjustmentFactors`** (see Table 3 below). For **`PolarizationAnalysis`** = `VectorPMD` the nonlinear terms are automatically corrected. If the factor 8/9 is already included in the measured value of the nonlinear refractive index, the value of the parameter **`NonLinearIndex`** (or the values of the nonlinear coefficient in the file **`NonlinearFilename`**) should be multiplied by a factor 9/8, or the parameter **`NonlinearAdjustmentFactors`** should be correspondingly modified.

The specification of $\gamma(f_k, f_i)$ quantities defined by formula (35) depends on the switch **`NonlinearDescription`**.

- **`NonlinearDescription`** = `NonlinearIndexParameter`: in this case nonlinear index n_2 is defined by parameter **`NonlinearIndex`**. Nonlinear coefficient $\gamma(f_k, f_i)$ is defined by (35).
- **`NonlinearDescription`** = `NonlinearFile`: nonlinear coefficients $\gamma(f_k, f_i) = \gamma(f_i)$ are read from the file defined by **`NonlinearFilename`**. The values responsible for XPM terms may be found using formula (35) by simple rescaling:

$$\gamma(f_k, f_i) = \gamma(f_k) \cdot \frac{f_i}{f_k} \cdot \frac{A_{eff}(f_k, f_k)}{A_{eff}(f_k, f_i)} \ . \tag{36}$$

All the terms responsible for the effects related to delayed Raman response are included into consideration of **`RamanScattering`** is set to `Yes`. If physical parameter **`RamanScattering`** is set to `No`, these effects are switched off irrespective of the states of the parameters **`XPM_MC`**, **`IntraBandRaman`**, **`InterBandRaman`**.

Similar to the power analysis, the Raman response function is read from the file defined by the parameter **`RamanFilename`**. The file may contain two (frequency/wavelength imaginary part of Raman response function) or three columns (frequency/wavelength imaginary + real part of Raman response function).

If the file contains the data for single pump wavelength, the formula (4) for imaginary part of response function $g(f_k, f_i) = \Im(G_R(f_k, f_i))$ is used. The real part is rescaled using a similar formula:

$$\gamma_R(f_k, f_i) = \rho\gamma(f_k, f_i)\Re[H(f_k - f_i)] = \frac{f_i}{f_p - |f_k - f_i|} \cdot \frac{A_{eff}(f_p, f_p - |f_k - f_i|)}{A_{eff}(f_k, f_i)} \cdot \Re[G_R(f_p, f_p - |f_k - f_i|)] \ . \tag{37}$$

If a third column is not specified, Raman response is assumed instantaneous and defined by:

$$\gamma_R(f_k, f_i) = \rho\gamma(f_k, f_i) \ , \tag{38}$$

where $\gamma(f_k, f_i)$ is defined by either (35) or (36) depending on the parameter **`NonlinearDescription`**.

If file **`RamanFilename`** contains two or more blocks, bilinear interpolation is applied for the real and imaginary parts of Raman response function. If the third column is not provided, the real part of Raman response function is defined by (38).

> **Note:** In the second column of Raman file the gain values (measured in 1/m/W) defined by (3), in the third column the values responsible for phase modulation (measured in rad/m/W) defined by (37) are specified. The gain is referred to the power gain, but we assume that the real part of Raman gain in measured relatively to the phase shift of the complex amplitude. As you can see, the factor of 2 is present in (3), but not in (37). But real part of the delayed response is usually found using Kramers-Kroenig relation using Hilbert transform of known imaginary part,
>
> that is, $$\Re(H(f)) = \frac{1}{\pi}\int_{-\infty}^{\infty} \Im(H(x))/(x - f)dx$$.
>
> This may be done using standard Matlab functions. In this case the restored real part will also contain the factor of 2. In order to avoid this problem, different units should be used in the Raman file. If the factor 2 is not considered

in the third column, [rad m^{-1} W^{-1}] units should be specified in the header line, if 2 is included, use [m^{-1} W^{-1}] units.

The polarization properties of nonlinear effects are provided by means of special adjustment factors defined by the parameter **`NonlinearAdjustmentFactors`**. These factors scale the strength of nonlinear interactions between copropagating bands, corresponding to terms in (31), (32) and (33). The meaning of the factors is described in Table 3.

Table 3 ***The meaning of the five adjustment factors entered in the parameter NonlinearAdjustmentFactors***

Nonlinear adjustment factor	This factor scales the strength of	Typical value
ξ_1	`SPM_EC`	8/9
ξ_2	`IntraBandRaman`	-
ξ_3	`XPM_EC`	2/3
ξ_4	`XPM_MC`	-
ξ_5	`InterBandRaman`	1/2

Please note that the nonlinear term is scaled using the Raman adjustment factors defined by the parameter **`RamanAdjustmentFactors`**. This is done for consistency with the power analysis.

Stimulated Brillouin Scattering

Brillouin scattering is considered only for separate samples of the band in the frequency domain, called Brillouin samples. The power of these samples decreases because of transition to corresponding counterpropagating distortions.

Rayleigh Scattering

The Rayleigh scattering does not appear explicitly in the field equations, since it generates counter-propagating waves. This is described by the generation of distortions in the power analysis and will be output from the module in the form of PSs. Furthermore, it is assumed that the attenuation due to Rayleigh scattering is already included into the attenuation constant. The power of the band reflected back and captured in the fiber core is passed to Rayleigh distortions.

Higher Order Nonlinear Effects

For ultrashort pulses (τ < 100 fs) it is necessary to includes higher-order nonlinear effects (**`HigherOrderNLEffects`** = `Yes`. In nonlinear operator (31)–(33), nonlinear coefficient γ_{ki} is assumed to be frequency-independent and considered for the center frequencies of interacting bands f_i and f_k. In general, the nonlinear operator in the frequency domain is proportional to the frequency f; moreover, A_{eff} in the denominator is also frequency-dependent.

Vector Analysis

Coarse Step Method

The propagation of light pulses can be described by the coupled nonlinear Schrödinger equations (1) or (2) from [3]. To find the solution of these equations it is necessary to make the computation steps much smaller than characteristic lengths. The typical fiber lengths (beat length Λ and correlation length *Lcorr*), which describe fiber birefringence, are usually less than the characteristic lengths for dispersion and nonlinearity. So it is computationally wasteful to use these small computation steps required to mimic the rapidly changing birefringence orientation.

UniversalFiber uses standard techniques for calculating polarization effects called "coarse step". In this method the continuous variations of birefringence are substituted by series of many short sections with constant birefringence. The axis of local birefringence varies from section to section. It means that the polarization state of the optical field is scattered to a new point on the Poincaré sphere as shown in Figure 10.

To avoid artificial periodicities in the spectrum of Stokes vectors, the length of each polarization scattering section, Δz_{scatt}, is randomly selected from a Gaussian distribution with mean and standard deviation, set by the parameters **`MeanStepWidth`** μ_{scatt} and **`WidthDeviation`** σ_{scatt}:

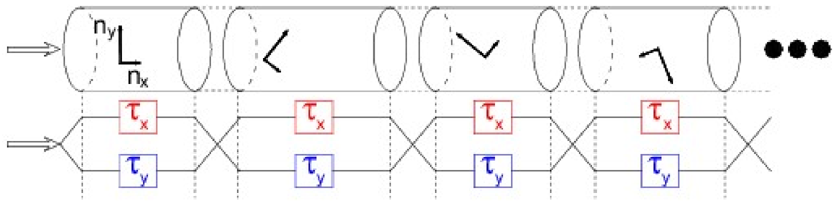


Figure 10 Schematic representation of the coarse step method

The parameter **WidthRandomNumberSeed** is used to initialize ('seed') the random number generator, which is used to find the scattering section length.

A realistic description of the polarization statistics is only possible if the number of "scattering" events is large. Therefore, the length of each scattering section is restricted to being much shorter than the fiber length, but longer than the fiber correlation length *Lcorr* defined by the parameter **CorrelationLength**.

We slightly modified the standard coarse step method by adding importance sampling (IS). This method allows treating statistical cases with larger DGD values more effectively than direct Monte-Carlo methods. It also allows calculating probabilities in the tails of the probability density function (pdf) because these events are very rare.

Each scattering section is defined by frequency-dependent Jones matrix U_j. Matrix U_j has the eigenvectors which coincide with the rotation axis in Stokes space. The multiplication of the input field by this matrix is equivalent to the rotation of polarization vector on the Poincaré around this rotation axis. The Jones matrix may be diagonalized, if its eigenvectors are known, that is, $U_j = T_j D_j T_j^{-1}$, where T_j is diagonalization matrix. The matrix $S_j = T_j^{-1} T_{j-1}$, which is constructed from the diagonalization matrices of subsequent sections, will be referred to as a "scattering matrix".

The rotation axes (local PMD vector) of the following section may be chosen randomly, or aligned close to the direction of PMD vector at the output of the previous section. In the last case the DGD value increases, because the length of PMD vector increases (see Figure 11). The PMD vectors of subsequent sections are calculated using PMD concatenation rule ((7.9) [11]):

$$\vec{\Omega}_{i+1} = \Delta\vec{\Omega}_{i+1} + R_{i+1}\vec{\Omega}_i \quad . \tag{40}$$

The angle between the rotation axis of the current section and total PMD vector at the end of the previous section is chosen in the following way ([27], [28]).

$$\theta = acos(2z^{1/\alpha} - 1) \quad , \tag{41}$$

where:

- z is a uniform random variable in [0,1],
- $\alpha \geq 1$ is a biasing factor, which is defined by the fiber parameter **CorrelateScatteringFactor** as $\alpha = 1/(1-$**CorrelationScatteringFactor**$)$. The value $\alpha = 1$ corresponds to the unbiased case, while increasing the value will bias the configuration toward increasingly large values of DGD. The angle φ which defines the position within the cone is randomly chosen in [0, 2π] interval. A geometrical interpretation of biasing is shown in Figure 11.

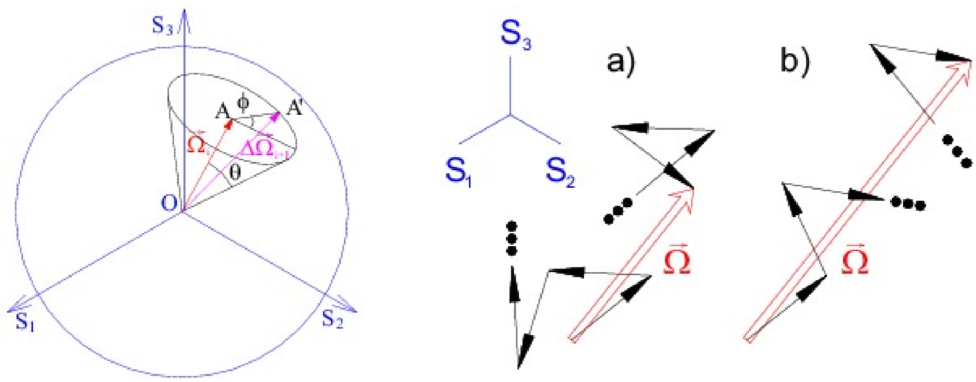


Figure 11 Geometrical interpretation of biasing procedure. Schematic representation of concatenation of PMD vectors of polarization sections: a) unbiasing case, b) biasing case

The maximum of pdf shifts to the larger DGDs too. The ratio of the probability of the case with normal conditions to that in the biased case is called likelihood ratio. The likelihood ratio of the whole fiber is defined as the product of likelihood ratios of the separate sections (we assume that these sections have approximately equal lengths).

The likelihood ratio of the separate sections is defined by

(42)

$$L_i = \frac{1}{\alpha}\left(\frac{2}{\cos\theta + 1}\right)^{\alpha - 1} .$$

The likelihood ratios may be stored to the file defined by the parameter **LikelihoodRatiosFilename**.

The pdf at specified point is defined as

$$P = \frac{1}{N}\sum_{k=1}^{N} I(\xi_k)L(\xi_k) \tag{43}$$

where:

- P is a probability of a certain event.
- N is a total number of trials.
- $L(\xi_k)$ is a likelihood ratio of the current trial.
- $I(\xi_k)$ is an indicator function that defines that the DGD value is located within a desired interval (bin).

The parameter **ScatteringRandomNumberSeed** is used to initialize ('seed') the random number generator used for the scattering matrices.

Nonlinear steps are included in an averaged form because the length of the section with constant birefringence is chosen to be significantly larger than the fiber correlation length *Lcorr*. The averaging procedure is described in detail in [3] and [4]. We have used an approximation of complete mixing of the polarization states on the Poincaré sphere (2π*Lcorr*>>Λ) and neglected the terms responsible for incomplete mixing, which may be omitted for typical regimes at which modern fibers operate. Similar to scalar case power analysis, equations are obtained from averaged field analysis equations using the representation of the bands in CW limit.

Vector Analysis: Fields

The nonlinear Schrödinger equation describing a propagation of the *i*-th sampled frequency band (SFB) with center frequency f_i and propagating in the forward direction is given by:

$$\frac{\partial}{\partial z}\vec{A}_i^+(t) = \underbrace{-\frac{\alpha}{2}\vec{A}_i^+(t)}_{attenuation} + \underbrace{\hat{D}_1\vec{A}_i^+(t)}_{PMD} + \underbrace{\hat{D}\vec{A}_i^+(t)}_{dispersion} + \underbrace{\hat{N}\vec{A}_i^+(t)}_{nonlinearity} , \tag{44}$$

where:

- $\vec{A}_i^+(t)$ is a column vector with elements $(A_{ix}^+(t), A_{iy}^+(t))^T$ which are the complex envelope for two polarization components for i channel (input signal) with center frequency f_i,
- N is a nonlinear operator that governs the effects of all the fiber nonlinearities on pulse propagation and its polarization properties.

Linear Operator

Unlike the scalar case, the linear operator includes the term responsible for linear PMD:

$$\hat{D}_1 = \beta_1\sigma_1\frac{d}{dt} = \beta_1\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\frac{d}{dt} , \tag{45}$$

The value of β_1 varies from one scattering section to the other and is defined by:

$$\beta_{1i} = \frac{D_{PMD}}{\sqrt{4\Delta z_{scatt,i}}} \tag{46}$$

where:

- β_1 is a specific group delay per unit length,
- D_{PMD} is a fiber PMD coefficient, defined by the parameter **PMDCoefficient**,
- $\Delta z_{scatt,\,i}$ is a current scattering step,
- σ_1 is a Pauli matrix.

The description of the terms responsible for attenuation and dispersion is similar to the scalar case.

Nonlinear Operator

The nonlinear operator in its simplest form is described by the Manakov term [3]

$$\hat{N}\vec{A}_i^+(z,t) = -j\gamma(f_i)\frac{8}{9}\left|\vec{A}_i^+(z,t)\right|^2\vec{A}_i^+(z,t) \quad (47)$$

This nonlinear term has been correspondingly extended to take into account the electronic and molecular contributions to the nonlinear interactions between different frequency bands, for the intra- and interband Raman scattering, and the higher-order nonlinear effects (see the parameters **`SPM_EC`**, **`XPM_EC`**, **`XPM_MC`**, **`IntraBandRaman`**, **`InterBandRaman`**, **`HigherOrderNLEffects`**).

The names of the nonlinear effects are chosen to be consistent with the scalar case:

- SPM_EC: the effects phase modulation due to instantaneous nonlinear contribution within one band for both the polarizations.
- XPM_EC: the effects phase modulation due to interactions with the bands whose index differ from the index of original band. This effect causes phase modulation and polarization rotation of the Stokes vector of i band around total Stokes vector of all other signals.
- Intraband Raman term is responsible for Raman nonlinear interactions within the i specified band, whereas Interband term for interactions between i and k bands, as well as with PSs and NB.

Raman Scattering

Raman gain strongly depends on the polarization of interactive waves. For example, copolarized gain is almost an order of magnitude larger than the orthogonal polarization gain near the peak [17], [18] (see Figure 12).

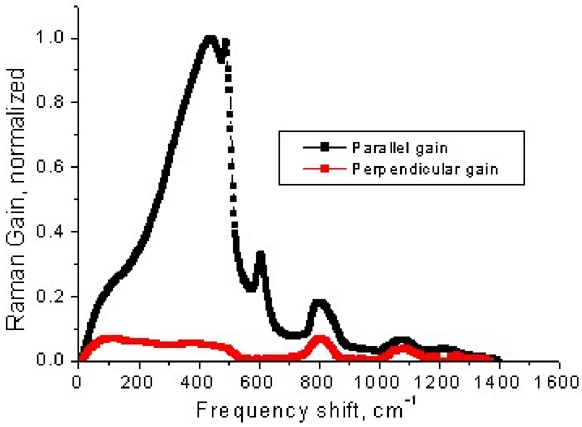


Figure 12 Normalized parallel and perpendicular Raman spectra

Raman gain profiles are read from the file defined by the parameter **`RamanFilename`**. The file may contain not only copolarized, but also transversal gain. If the data for transversal gain is not provided, we assume that the strength of the effect is very small and can be neglected.

The second column contains an imaginary part of the copolarized Raman profile responsible for the gain, the third column (if specified) contains the real part of the copolarized Raman profile responsible for phase polarization effects. Its values are defined by the following formula:

$$g_{//}(f_k, f_i) = 2\rho\gamma(f_k, f_i)H_{//}(f_k - f_i) \ , \quad (48)$$

where:

- $H_{//}(f_k - f_i)$ is a normalized copolarized Raman response function defined by:

$$H_{//}(\Delta f) = \int_0^{\infty}[\alpha(\tau) + \beta(\tau)]e^{j2\pi\Delta f\tau}d\tau \ . \quad (49)$$

- $\gamma(f_k, f_i)$ is a fiber nonlinear coefficient (35).

The data for transversal Raman profile may be specified in the fourth and fifth columns.

$$g_{\perp}(f_k, f_i) = 2\rho\gamma(f_k, f_i)H_{\perp}(f_k - f_i) \ , \quad (50)$$

where:

- $H\perp(f_k - f_i)$ is a normalized transversal Raman response function defined by:

$$H_{\perp}(\Delta f) = \frac{1}{2}\int_0^{\infty} \beta(\tau)e^{j2\pi\Delta f\tau}d\tau \quad . \tag{51}$$

The file may contain either one block of data, in which case a Raman profile for one pump frequency is specified, or many blocks for many pump frequencies. In the first case specified data are rescaled, in the second, bilinear interpolation is used. The scaling due to Raman gain frequency dependence is similar to the scalar case (see (4) and (37)).

For a detailed description of the Raman file format, see Appendix.

Stimulated Brillouin Scattering

We assume that only the copolarized component of the Brillouin distortion contributes to the signal power degradation [24].

Numerical Parameters

In this section:

This section describes the numerical algorithms used in the power and field analysis and explains how to set the numerical parameters in order to control the accuracy of computations and to resolve possible convergence problems. First, the numerical algorithm for the power analysis is described. Then the numerical parameters that control the accuracy of the field analysis are explained.

Power Analysis

This section describes the numerical algorithms used in the power analysis and explains how to set the numerical parameters in order to control the accuracy of computations and to resolve possible convergence problems.

The goal of the numerical algorithm used for the power analysis is to solve the two-point boundary value problem (2PBVP) with the accuracy set by the parameter `AccuracyGoal`. The 2PBVP is solved using the iterative algorithm. The iterations are continued until the difference in signal powers obtained at two consecutive iterations becomes less than the desired accuracy.

However, the actual obtained accuracy is not defined solely by the number of iterations because the accuracy also depends on the step size used for ODE solution. In order to ensure that the obtained relative error does not exceed the `AccuracyGoal`, the algorithm chooses the step size of the computational grid by itself. First, the computation grid for ODE integration is taken so that the step size is equal to `InitialStepSize`, and the iterations of the 2PBVP solver are performed using this grid. (Iterations on a grid with a constant step size will be called the "internal iterations" below.)

The parameters that control the numerical algorithm for the power analysis are the `AccuracyGoal`, `IterationAccuracyFactor`, `StabilizationFactor`, `MaximumIterations`, `InitialStepSize`, and `MinimumStepSize`.

- `AccuracyGoal` specifies the target accuracy of the solution of the 2PBVP. The algorithm chooses the computation grid and the number of iterations that are necessary to achieve the `AccuracyGoal`. The `AccuracyGoal` is specified in dB.

  Table 4 contains relative accuracy factors that correspond to the specified `AccuracyGoal` parameter.

  Table 4 ***Relative accuracy factors corresponding to the AccuracyGoal parameter***

AccuracyGoal	Relative accuracy factor
0.01	2.3e−3
0.001	2.3e−4
0.0001	2.3e−5
...	...

  This relative factor defines a measure of the error, that is, the maximum difference between the solutions for consecutive grids. If the obtained error is less than this factor, the desired solution is achieved and iterative algorithm stops.

- **IterationAccuracyFactor** controls the accuracy of the internal iterations with constant step size. The internal iterations are continued until the difference between the two consecutive iterations becomes less than the accuracy defined by the parameter **AccuracyGoal** divided by **IterationAccuracyFactor**. For example, if **IterationAccuracyFactor** = 10, the accuracy that must be obtained during the internal iterations is 10 times better than the overall accuracy **AccuracyGoal**.

  In most cases the algorithm works without any problems with default setting of **IterationAccuracyFactor**. However, in some complicated cases the internal iterations converge towards the true solution very slowly. In such cases the comparison of solutions for different grids does not make sense because these solutions are not sufficiently accurate. The algorithm may reduce the step size down to a very small value, and yet it will not obtain a reliable solution. Problems of this kind can be avoided by increasing the **IterationAccuracyFactor** up to 100 or larger values. On the other hand, in more typical setups in which the internal iterations converge very quickly, decreasing this parameter can slightly reduce the overall computation time.

  Please note that the plot of the relative error versus the iteration number, which is useful for examining the convergence of the algorithm, can be visualized by setting the **Convergence** parameter to On.

- **MaximumIterations** specifies the maximum number of internal iterations that can be done by the algorithm. If the **AccuracyGoal** is not achieved after the maximum number of iterations, an error message is issued. This parameter is useful for limiting the computation time. For example, if the iterative algorithm gets into a limit cycle, the simulation will not continue forever but will stop after the specified number of iterations. If the iterative algorithm converges slowly, for example, if a small value of **StabilizationFactor** is used, it might be necessary to increase the **MaximumIterations** in order to allow the algorithm to achieve the required accuracy.

  Please note that the maximum computational time for a given schematic is not necessarily proportional to **MaximumIterations**. For example, doubling **MaximumIterations** may lead to a fourfold increase in the maximum computational time.

- **InitialStepSize** corresponds to an actual computation step along the length of the fiber. Please note that the accuracy of the solution is determined by the parameter **AccuracyGoal** and does not depend on the initial step size. Also note that decreasing this parameter often increases the simulation time.

  In case of a multispan fiber, **InitialStepSize** can contain either one value, or **NumberOfFiberSpans** values. If one value is entered, the initial grid has the same step size for all fiber spans. Entering the initial step sizes for each fiber span can be useful if physical properties of the fiber spans are significantly different. For example, if the attenuation of some fiber span is much larger than that of the others, it is useful to set the initial step size for this span smaller than for the other spans, because it can lead to reduced computation time. In practice, it is recommended to run the simulation with the default value of **InitialStepSize** = 1000 (that is, 1 km), and then try to speed up the simulation by changing the initial step sizes for the fiber spans with different physical properties.

- The **MinimumStepSize** parameter specifies the smallest possible step size of the solver. Similar to the **MaximumIterations** parameter, **MinimumStepSize** also limits the maximum simulation time by not allowing the algorithm to use an excessively small step size. An error message is issued if the **MinimumStepSize** has been reached on at least one fiber span, and in this case the problem can be solved either by decreasing the **MinimumStepSize**, or increasing the **IterationAccuracyFactor** (see the description of the IterationAccuracyFactor). The following approach can be recommended if the algorithm is stopped because of the MinimumStepSize limitation: first, the IterationAccuracyFactor should be multiplied by 2 or 4, and if the problem persists, the MinimumStepSize should be divided by 2. This process can be repeated until the problem disappears.

- **StabilizationFactor** adjusts the rate of convergence of the internal iterations. If the stabilization factor is set to 1.0, the algorithm uses no stabilization. If smaller values of this parameter are used, the algorithm becomes more stable.

  This parameter can be used for solving convergence problems that occur in some complicated setups. In such setups the iterative algorithm can get into some kind of limit cycle, which can be observed in the convergence plots, which will contain jumps and oscillations. The iterations do not converge to a solution and an error message that the **MaximumIterations** is exceeded will be issued. Such problems can be resolved by decreasing the stabilization factor, sometimes down to 0.1 or even smaller.

  > **Note:** When Brillouin scattering is considered (**BrillouinScattering** = Yes), the stabilization factor should be set to a rather small value (approx. equal to 0.2), because Brillouin gain is three orders of magnitude larger than Raman gain, which may cause problems with the convergence for 2PBVP.

In many cases, it is not necessary to solve 2PBVP. The typical example is when the signals propagate in one direction, or when the signals propagated in different directions do not affect to each other. It means that the propagation equations may be quickly integrated. The parameter **BidirectionalAnalysis** controls whether the power analysis differential equations are solved using the 2PBVP algorithm or the equations are simply integrated in corresponding directions. When signals propagate in the fiber in only one direction, setting this parameter to Simplified can significantly speed up (10–100 times) the power analysis. However, if signals propagate in both directions, the interactions between counterpropagating signals are not considered in the Simplified mode. It means that the effects of Raman scattering of counterpropagating signals, Rayleigh scattering, Brillouin scattering, and bidirectional noise generation cannot be included. In this case an error message is issued. If this parameter is set to Full, all interactions between counterpropagating signals are considered and all the effects can be included.

When Four-Wave Mixing is simulated with bidirectional power analysis (**FWM** = On, **BidirectionalAnalysis** = Full), the module requires a sufficiently small step size to ensure convergence of 2PBVP. In this case, the value of the

FWMPowerSeed parameter should be increased to keep the number of simulation times reasonably small.

The descriptions of the parameters **AccuracyGoal**, **IterationAccuracyFactor**, **StabilizationFactor**, **MaximumIterations**, **InitialStepSize**, and **MinimumStepSize** written above are related to the case when 2PBVP is solved. Some of them are not used when simplified analysis is carried out.

The input parameters for the integrator are:

- Relative accuracy parameter passed to ODE solver ε (see [29] for details): its default value is usually set to 1.0e−6. It is defined by the parameters **AccuracyGoal** and **IterationAccuracyFactor** in the following way: $\varepsilon = (1-10^{-\text{AccuracyGoal}/10})$**/IterationAccuracyFactor.** For example, if ε approximately equals to 1.0e-6, **AccuracyGoal** and **IterationAccuracyFactor** may be set to 0.0001 and 10 correspondingly.
- Initial step size parameter passed to ODE solver: defined by the parameter **InitialStepSize**.
- Minimum step size passed to ODE solver: defined by the parameter **MinimumStepSize**. If integration step becomes smaller than this parameter, an error message is generated.

Numerical algorithms for the power analysis in vector case are similar to the ones for scalar, but the convergence is checked not only for the powers, but also for the nonnormalized components of Stokes vectors. Note that the components of the Stokes vectors converge much more slowly than the powers, because they are rotated due to linear PMD and fiber nonlinearities. This may slow down the convergence significantly. If the frequencies of the signals are widely separated, their polarizations mix rapidly, so the strength of the interactions depends on powers only, and the averaged gain proportional to polarization-dependent terms is negligible. So it is quite reasonable to specify the frequency range in which polarization properties of nonlinear effects are considered. The parameter **NonlinearPMDBandwidth** defines the bandwidth around any signal frequency in which nonlinear polarization-dependent interactions take place. Outside this range we assume that polarization-dependent interactions are ignored.

The convergence plots can be visualized by setting **ConvergencePower** to `On`. In the vector case, the convergence plot may be visualized for the Stokes vectors components having the worst convergence (**ConvergenceStokes** = `On` used jointly with **ConvergencePower** = `On`).

In *UniversalFiber* there are two parameters that exclude the terms whose contribution is too small, and so it may be neglected. These interactions are usually referred to the noise and distortions. This approximation is usually acceptable when the noise/distortion power is much smaller than the signal power, and can speed up the computation significantly. If the enhanced parameter **NoiseBinInteractions** is set to `No`, then the Raman power transfer (+nonlinear polarization rotation in vector case) from noise bins to the other signals is neglected. We recommend using this switch in preliminary simulations. Parameter **DistortionInteractions** has the same meaning, but excludes the interactions of the distortions with other signals.

Field Analysis

Nonlinear Schrödinger equations are solved using the split-step Fourier method, which allows to treat the linear and nonlinear operators in different domains [1]. In *UniversalFiber* we used two types of split-step algorithm: asymmetric and symmetric. In the former case, at the first step the nonlinearity acts alone (that is, D = 0), at the second step, dispersion acts alone (N = 0). In the later case, nonlinearity is switched on at the middle of the segment *dz* (see Figure 13). The symmetric one provides higher accuracy at little extra effort, so we recommend that you use this type. Parameter **SplitStepType** switches between two computation regimes.

The accuracy of the field analysis is controlled by several parameters as follows:

- If the **StepSelectionMethod** parameter is set to `LocalErrorMethod`, the step is chosen adaptively during the integration along the fiber according to the algorithm described in [10], maintaining the relative error at each step within range [ε/2; ε], where the desired error value ε is specified in the parameter **TargetErrorValue**. This mode is rather similar to adaptive step algorithms routinely used in ordinary differential equations.

  When using the local error method, the following should be observed:

  + Local error method has greater order with respect to step size than the other options available for the fiber modules, which means that it will be more efficient (require fewer steps) for high-accuracy simulations. However, due to a larger number of computations per step, low-accuracy simulations can be somewhat longer.
  + In some cases, such as propagation of several frequency components in a shifted-dispersion fiber, the local error method tends to make larger errors in low-power regions of the spectrum (like the FWM products within a sampled band).

  Note that the local error method implicitly uses a *symmetrical* split-step algorithm (see details in [10]).

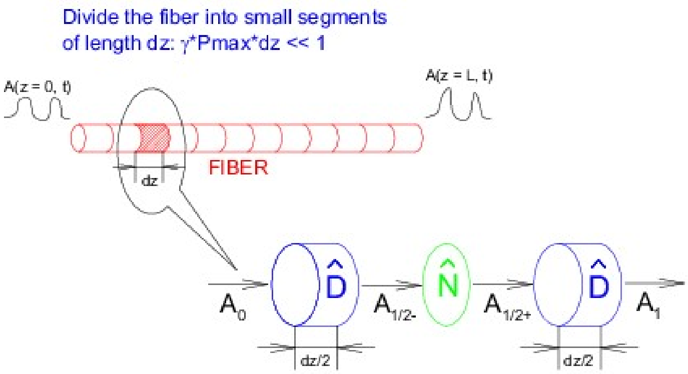


Figure 13 Symmetric split-step algorithm

The parameter **MaxPhaseChange** is very important because it defines the final accuracy and also the time required for simulation. For example, the default value of 0.05 degrees may be too small for your schematic and it can be increased to 0.5 degrees without noticeable accuracy loss. However, the field analysis speed will increase by 10!

Note: For both local error and nonlinear phase change methods, the step size is limited from above and below by the **MaxStepWidth** and **MinStepWidth** parameters, respectively.

If the desired step size is less than **MinStepWidth**, the module will stop with an error message. Normally, the minimal step size is set to ensure that the simulation will not take an excessive amount of time.

The parameter **MaxStepWidth** sets the upper limit of the step size so that steps larger than **MaxStepWidth** cannot be used. If the step size found using the **MaxPhaseChange** parameter is larger than **MaxStepWidth**, a step of **MaxStepWidth** will be performed. Decreasing **MaxStepWidth** is useful for avoiding excessively long steps and significant numerical errors possible during these steps.

- If **StepSelectionMethod** is set to `ConstantStep`, the constant step size equal to he value of the **MaxStepWidth** parameter will be used.
  This setting can be useful if you wish to make the simulation time independent of the SFB/MFB power.

PMD Modeling

There are many parameters which control the generation of the random scattering matrices when vector analysis is considered (**PolarizationAnalysis** = `VectorPMD`).

The two parameters (**WidthRandomNumberSeed** and **ScatteringRandomNumberSeed**) defining the "seed" for random generators are described in the coarse step section. The sequence of scattering section lengths and scattering matrices along the fiber can be retained for multiple input blocks of data, using the parameters **FreezeWidth** and **FreezeScattering**. The value of **FreezeWidth** sets the number of particles (input blocks) for which the same scattering section lengths are used, while **FreezeScattering** similarly controls the scattering matrices. This behavior is followed for multiple passes through the fiber in a single iteration (such as loop experiments), or for multiple runs of a simulation. Note that for both these parameters, values of 0 or 1 have the same functionality, causing new values to be generated for each block. For PMD analysis of the fiber using the *TestSetJonesMatrix*, **FreezeWidth** and **FreezeScattering** should both be set to 2.

The module may write/read the scattering positions and scattering matrices/rotation axes to/from the file. Parameter **BirefringenceDataType** defines which type or data should be stored: it may be Rotation matrices (**BirefringenceDataType** = `RotationMatrices`), that is, unitary scattering matrices at scattering points or rotation axes of the sections (**BirefringenceDataType** = `RotationAxes`). Parameter **BirefringenceProfile** controls whether the data is stored in or retrieved from the file. There are three possible values:

- `None`: scattering positions and matrices are calculated using the fiber parameters.
- `Store`: scattering positions and matrices/rotation axes will be written to the file defined by **BirefringenceOutputFilename**.
- `Retrieve`: scattering positions and matrices/rotation axes are read from the file defined by **BirefringenceInputFilename**.

The first line of these files defines the type of data. It is either:

```
# BirefringenceDataType = RotationAxes, or
# BirefringenceDataType = RotationMatrices.
```

The next four lines give the parameters **MeanStepWidth**, **WidthDeviation**, **WidthRandomNumberSeed**, **ScatteringRandomNumberSeed**:

```
# MeanStepWidth = 1000
# WidthDeviation = 100
# WidthRandomNumberSeed = 5
```

```
# ScatteringRandomNumberSeed = 5
```

The sixth line differs for the two cases:

- **`BirefringenceDataType`** = `RotationAxes`
  `# StartPos EndPos r1 r2 r3 LikelihoodRatio fiAtCarrierFrequency`
  where:
  + `StartPos` is the start position of the section,
  + `EndPos` is the end position of the section,
  + `r1 r2 r3` are rotation axis of the current section in Stokes space,
  + `LikelihoodRatio` is a likelihood ratio of the current section,
  + `fiAtCarrierFrequency` is a retardation angle at carrier frequency chosen to be random number between 0 and 2п.
- **`BirefringenceDataType`** = `RotationMatrices`
  `# ScatteringPosition Re(J00) Im(J00) Re(J10) Im(J10) LikelihoodRatio`
  where:
  + `ScatteringPosition` is a current scattering position,
  + `Re(J00) Im(J00) Re(J10) Im(J10)` are the real and imaginary parts of corresponding elements of the scattering matrix. It should be noted that two other Jones (unitary) matrix elements can be calculated from the elements given, using the following formulae:
    Re(J01) = −Re(J10), Im(J01) = Im(J10), Re(J11) = Re(J00), and Im(J11) = −Im(J00).
  + `LikelihoodRatio` is a likelihood ratio of current section.

In both cases, the remaining lines contain the numerical data corresponding to the specified column. The likelihood ratio of the current fiber section is written in the last column.

The last line contains the total likelihood ratio of the whole fiber:

```
# Overall Likelihood Ratio = 5.8954254512e-001
```

The file format for the birefringence files in *UniversalFiber* module defined by **`BirefringenceInputFilename`** and **`BirefringenceOutputFilename`** for the case of single span are backward compatible with the file formats for *FiberNLS_PMD*.

> **Note:** The format of the birefringence files in the case of a number of spans is similar to a single span case, but the keyword `# Span ...` is added before the main block.

A file with calculated likelihood ratios for different runs defined by the parameter **`LikelihoodRatiosFilename`**. The first line contains the header:

```
# RunNumber OverallLikelihoodRatio
```

The following lines are the corresponding run numbers and likelihood ratios, for example

1 `0.858e-1`

2 `0.984e-1`

... and so on. If several fiber spans are simulated (**`NumberOfFiberSpans`**>1), the stored value of the likelihood ratio is the product of likelihood ratios of all fiber spans.

If the **`FreezeScattering`** parameter is greater than 1, the likelihood ratio is written only once per **`FreezeScattering`** runs. The file **`LikelihoodRatiosFilename`** works in "append" mode, that is, the data is added to the file end.

Parameterized Signals

The Signal Statistic attributes of parameterized signals are changed:

Fiber length is added to the value of *Accumulated Fiber Length* attribute. The phase shift φ_{NL} due to self-phase modulation is calculated [30]:

$$\phi_{NL} = \gamma P_{in} L_{eff} \qquad (52)$$

where P_{in} is the input signal power, L_{eff} is the effective fiber length. Then it is added to the value of the *AccumulatedSPM* attribute. The value σ_T of the *AccumulatedDGD* attribute after the fiber is [30]

$$\sigma_T = \sqrt{\sigma_{Tin} + D_{PMD}^2 L} \qquad (53)$$

where σ_{Tin} is the value of the input signal *AccumulatedDGD*, L is the fiber length, D_{PMD} is the fiber PMD coefficient.

The dispersion accumulated in the fiber is added to the signal attribute *AccumulatedDispersion*.

For parameterized signals, the values of a number of physical parameters are saved to the Tracking Data (see Chapter 7 of the *VPItransmissionMaker™Optical Systems User's Manual*). They can be visualized using the

LinkAnalyzer module. The fiber parameter **GroupRefractiveIndex** is used to calculate the fiber delay (**TransitTime** in *LinkAnalyzer*).

Debug and Visualization

A number of parameters in the Visualization category are used to control internally calculated quantities that can be stored in a file specified by the **LogFilename** parameter, and/or displayed in visualizer windows. These quantities may be the power/Stokes vector distributions along the fiber and also the convergence of the iterative boundary value problem solver used in the power analysis.

If the parameter **VisualizationMode** is set to `StartAutomatically` or `StartOnDemand`, the power and/or Stokes vector component distributions for the following kinds of signals can be visualized:

- Input parameterized signals, parameterized signals derived from the sampled bands during discretization at the power analysis stage (see Discretizer) and parameterized signals derived from the sampled bands when the Brillouin scattering effect is included (see Brillouin Distortions) — if the parameter **Signals** is set to `On`. On the visualizer, the signals of the first and third types are labeled as `Sig`, the signals of the second type are labeled as `Sig*`.
- Noise bins, if the parameter **NoiseBins** is set to `TotalPower` or `IndividualPowers` (the total power of all noise bins or the power distribution of each noise bin, correspondingly).
- Any kind of distortions, if the parameter **Distortions** is set to `On`. Additionally, each kind of distortions can be switched on/off individually via the corresponding parameters: **CrosstalkDistortions**, **RayleighDistortions**, **BrillouinDistortions**, **FWMDistortions**, or **GenericDistortions**.

The parameter **SignalDirections** can be used to visualize only forward-, only backward-, or both forward- and backward-propagating powers.

If the parameter **StokesVector** is set to `On` or `Normalized`, the nonnormalized or normalized components of the Stokes vectors of the signals, or noise, or distortions must be visualized.

If **SaveToFile** is set to `On` and the valid file name is specified via the **LogFilename** parameter, the power/Stokes vector distributions are saved to the file. Only those power/Stokes vector distributions are saved to file whose corresponding parameters are not set to `Off`.

The **ConvergencePower** indicates that the convergence of the boundary value problem solver for the power analysis must be visualized. The visualizer window will show the relative error versus the iteration number. This information can be useful for choosing the optimum values of the numerical parameters of the module. The parameter **ConvergenceStokes** indicates the convergence curve for the component of the Stokes vector which has the worst convergence (plotted in a new window). This parameter can be used when performing vector polarization analysis (that is, **PolarizationAnalysis** = `VectorPMD`).

The parameter **ConvergenceGrid** indicates that the convergence plot for the grids with different step size must be shown. Each point on the plot defines the relative error between the solutions at two consecutive grids. The number of the points on this plot equals the number of grids (minus one) that is necessary to achieve the solution specified by **AccuracyGoal** parameter. The last two parameters are used only jointly with the **ConvergencePower** parameter.

The enhanced parameter **PlotFrequencyRange** defines the frequency interval(s) of the signals to be visualized on the plots. The power distributions will be visualized (or saved to file) only for signals with frequencies in the specified range(s). One or several frequency ranges can be specified, like `(190.0e12, 191.0e12) (193.0e12, 194.0e12)`. The upper and lower ranges should be separated by the comma and enclosed in parenthesis. If the lower frequency is greater than the upper one, they trade places. This is very convenient when these frequencies are specified by means of known wavelengths, because frequency is inversely proportional to wavelength.

> **Note:** The expressions may also be used to specify plot ranges. For instance, let us assume that we have a schematic parameter C which defines the speed of light. The ranges may be specified in the following way: `(!” expr {C}/1540e-9 “, !” expr {C}/1610e-9 “)`.

The parameter **PowerPlotPoints** specifies the number of points to be used for visualization of the power/Stokes vector distributions along the fiber. It also defines the number of the points to be stored to the file. When vector analysis is carried out, the number of the plot points is greater than **PowerPlotPoints**, because the powers and the components of the Stokes vector are visualized before and after the scattering points. Note that powers at event points are also visualized twice: before and after event points.

Random Number Generators

The noise is generated during the propagation of the signals in the fiber. In VPI Design Suite, noise may be considered separately or incorporated into the sampled band. The global parameter **InBandNoiseBins** controls this behavior: if it is set to `OFF`, generated noise will be added to the sampled bands, where they exist, as random fields (and will be added to noise bins elsewhere in the spectrum). If this parameter is set to `ON`, the noise is kept as the noise bins separately. Noise is added to the sampled bands with the random phase. Parameter **NoiseRandomNumberSeed** is the random seed lookup index which is used for incorporating noise bins into sampled bands (if the Global parameter **InBandNoiseBins** = `OFF`). This parameter is an array that may contain one or two values. If two values are provided, then the first is the lookup index used for forward-propagating noise, and the second is the lookup index for backward-propagating noise.

If only one value is provided, then it is used for both forward- and backward-propagating noise. In either case, a value of zero causes a unique seed to be automatically selected. This parameter is needed to obtain similar outputs after the runs with fixed parameters.

Reinitialization

Reinitialization is controlled by the parameters **`FreezeWidth`** and **`FreezeScattering`**. If these parameters are set to 0 or 1, then their random number generators are not reset, that is, the noise sequence for randomly changing birefringence is continued. The value of **`FreezeWidth`** sets the number of runs for which the same scattering section lengths are used, while **`FreezeScattering`** similarly controls the scattering matrices.

> **Note:** The scattering matrices can be kept for the same or a smaller number of module runs as the section lengths, as newly generated random values of lengths generally change the number of scattering matrices. In other words, a **`FreezeScattering`** parameter value greater than a **`FreezeWidth`** parameter value is ignored.

These parameters are used to simulate the same scattering behavior on 2 or several runs, as it is necessary for module *TestSetJonesMatrix*.

The parameter **`Active`** enables or disables module execution. If **`Active`** = `Off`, then all input signals are passed to outputs unaltered.

References

[1] G.P. Agrawal, *Nonlinear Fiber Optics*, Second edition, Academic Press, 1995.
[2] S.A. Akhmanov, S.A. Vysloukh, A.S. Chirkin, *Optics of Femtosecond Laser Pulses*, American Institute of Physics, 1991.
[3] D. Marcuse, C. R. Menyuk, P. K. A. Wai, "Application of the Manakov−PMD equation to studies of signal propagation in optical fibers with randomly varying birefringence," *Journal of Lightwave Technology*, vol. 15, no. 9, September 1997, pp. 1735−1746.
[4] P. K. A. Wai et al., "Nonlinear polarization-mode dispersion in optical fibers with randomly varying birefringence," *J. Opt. Soc, Am. B*, Vol. 14, No. 11, November 1997, pp. 2967−2978.
[5] E.A. Glolovchenko, A.N. Pilipetskii, "Unified analysis of four-photon mixing, modulational instability, and stimulated Raman scattering under various polarization conditions in fibers," *J. Opt. Soc, Am. B*, vol. 11, no. 1, January 1994, pp. 92−100.
[6] C.R. Menyuk, M.N. Islam, J.P. Gordon, "Raman effects in birefringent optical fibers," *Optics Letters*, vol. 16, no. 8, April 15, 1991, pp. 566−568.
[7] D. Wang, C.R. Menyuk, "Polarization evolution due to the Kerr nonlinearity and chromatic dispersion," *Journal of Lightwave Technology*, vol. 17, no. 12, December 1999, pp. 2520-2529.
[8] D. Wang, C.R. Menyuk, "Reduced model of the evolution of the polarization states in wavelength-division-multiplexed channels," *Optics Letters*, Vol. 23, No. 21, November 1, 1998, pp. 1677–1679.
[9] C.R. Menyuk, "Application of multiple-length-scale methods to the study of optical fiber transmission," *Journal of Engineering Mathematics* 36: 113–136, 1999.
[10] O. V. Sinkin, R. Holzlöhner, J. Zweck, C.R. Menyuk, "Optimization of the split-step Fourier method in modeling optical-fiber communications systems," *J. Lightwave Technol.,* vol. 21, no. 1, pp. 61–68, January 2003.
[11] J. P. Gordon, H. Kogelnik, "PMD fundamentals: Polarization mode dispersion in optical fibers," Review, *PNAS*, April 25, 2000, vol. 97, no. 9, pp. 4541–4550.
[12] "Definition and test methods for the relevant parameters of single-mode fibres," *ITU-T Recommendation G.650*.
[13] H. Kidorf, K. Rottwitt, M. Nissov, M. Ma, and E. Rabarijaona, "Pump Interactions in a 100-nm bandwidth Raman amplifier," *IEEE Photonics Technology Letters*, vol.11, no. 5, May 1999, pp. 530–532.
[14] Ch. Headley III, G.P. Agrawal, "Noise Characteristics and Statistics of Picosecond Stokes Pulses Generated in Optical Fibers Through Stimulated Raman Scattering," *IEEE Journal of Quantum Electronics*, vol. 31, no. 11, May 1995, pp. 2058–2067.
[15] N.R. Newbury, "Pump-wavelength dependence of Raman gain," *OAA 2002, OME9*.
[16] A. Berntson, S. Popov, E. Vanin, G. Jacobsen and J. Karlsson "Polarization dependence and gain tilt of Raman amplifiers for WDM systems," *OFC 2000, MI2−1*.
[17] M. N. Islam, "Raman Amplifiers for Telecommunications", *IEEE Journal of Quantum Electronics*, vol. 8, no. 3, May/June 2002, pp. 548–559.
[18] R.H. Stolen, C. Lee, and R.K. Jain, "Development of the stimulated Raman spectrum in single-mode fibers," JOSA B, vol. 1, no 4, 1984, pp. 652-657.
[19] D. Derickson, *Fiber Optics. Test and Measurement,* Prentice Hall, 1998.
[20] Optical Time Domain Reflectometer (OTDR) Data Format, *Special Report SR-4731*. Telcordia, Morristown, NJ, January 2000. http://www.telcordia.com/
[21] *Corning Single-Mode Optical Fiber, SMF-28 fiber*, product information. http://www.corning.com/opticalfiber/
[22] E. Lichtman, R.G. Waarts, A.A. Friesem, "Stimulated Brillouin scattering excited by a modulated pump wave in single-mode fibers," *Journal of Lightwave Technology*, vol. 7, no. 1, January 1989, pp. 171−174.
[23] A. Djupsjöbacka, G. Jacobsen, and B. Tromborg, "Dynamic stimulated Brillouin scattering analysis," *Journal of Lightwave Technology*, vol. 18, no. 3, March 2000.
[24] R. Stolen, "Polarization in Raman and Brillouin lasers," *IEEE Journal of Quantum Electronics*, vol. QE-15, no. 10, October 1979, pp. 1157–1160.
[25] Q. Lin, G.P. Agrawal, "Statistics of polarization-dependent gain in fiber-based Raman amplifiers," *Optics Letters*, vol. 28, no. 4, February 15, 2003, pp. 227-229.
[26] Q. Lin, G.P. Agrawal, "Polarization mode dispersion–induced fluctuations during Raman amplifications in optical fibers," *Optics Letters*, vol. 27, no. 24, December 15, 2002, pp. 2194−2196.

[27] G. Biondini, W. L. Kath, and C. R. Menyuk, "Importance sampling for polarization-mode dispersion," *IEEE Photonics Technology Letters*, vol. 14, no. 2, pp. 310–312, February 2002.
[28] S. Fogal, G. Biondini, W. L. Kath, "Importance sampling for polarization mode dispersion," *Venice Summer School on Polarization Mode Dispersion*, June 24–26, 2002.
[29] W.H. Press et al., *Numerical Recipes in C*, Cambridge University Press, second edition.
[30] G.P. Agrawal, *Fiber-Optic Communication Systems*, 2nd edition. Wiley Interscience, 1997, pp. 46-63.

Appendix

In this section:

Loading Pre-Saved Fiber Parameters

The characteristics of several fiber types commonly used in real applications (fiber profiles) can be loaded into modules from predefined .vtmp files.

The characteristics of the following fiber types are available:

- DCF (Dispersion Compensation Fiber): the data are based on measurements and product information of the Corning PureForm SMF DCM-40 Module. Attenuation and dispersion file measurements provided by Acreo AB, Photonics Department;
- DSF (Dispersion Shifted Fiber): the data are based on measurements and product information of Corning SMF-DS Fiber. Attenuation and dispersion file measurements provided by Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institut, Department of Photonic Networks and Systems;
- HighBitRate (HighBitRate Fiber): the data are based on measurements and product information of Draka TeraLight Ultra Optical Fiber. Attenuation and dispersion file measurements provided by Fraunhofer-Institute for Telecommunications, Heinrich-Hertz-Institut, Department of Photonic Networks and Systems;
- LWP (Low Water Peak Fiber): the data are based on measurements and product information of Corning SMF-28e Fiber;
- Metro (Metropolitan Network Fiber): the data are based on measurements and product information of Corning MetroCor Fiber;
- NZ-DSF_LargeEffArea (NZ-DSF Large Effective Area fiber): the data are based on measurements and product information of Corning LEAF Fiber. Attenuation and dispersion file measurements provided by Fraunhofer-Institute for Telecommunications, Heinrich-Hertz-Institut, Department of Photonic Networks and Systems;
- SMF (Standard Fiber): the data are based on measurements and product information of Corning SMF-28 Fiber. Attenuation and dispersion measurements provided by Acreo AB, Photonics Department;
- Submarine (Submarine Fiber): the data are based on product information of Corning Vascade Fiber;
- ZWP (Zero Water Peak Fiber): the data are based on measurements and product information of OFS AllWave FLEX ZWP Fiber. Attenuation and dispersion file measurements provided by Fraunhofer-Institute for Telecommunications, Heinrich-Hertz-Institut, Department of Photonic Networks and Systems.

To load a fiber profile from a .vtmp file, open the **Parameter Editor**, click **Load parameter settings** (down arrow icon) and select the desired profile from the list.

Import of OTDR Data

Information about the spatial distribution of attenuation, reflections, and backscattering characteristics of optical fibers can be obtained using OTDR (Optical Time Domain Reflectometer) measurements [20]. This information may be used to specify fiber parameters, such as **`Attenuation`**, **`RayleighBackscatterCoefficient`**, **`EventLoss`**, and **`EventReflectance`** as well as to specify frequency dependence of these characteristics using results of OTDR measurements with different wavelength sources. These characteristics can be imported directly from OTDR data file(s) specified using the parameters **`OTDRForwardFilename`** and/or **`OTDRBackwardFilename`**. Switch **`UseOTDRdata`** specifies whether or not the characteristics of a multispan fiber link should be imported from these files. The general requirements for imported OTDR files are the following:

- Results of OTDR measurements must be saved in accordance with Telcordia format SR-4731 [20].
- OTDR files must contain the Key Event block.
- The number of Key Event fields in each OTDR file must be equal to **`NumberOfFiberSpans`** plus one.
- The OTDR files specified by parameters **`OTDRForwardFileName`** and **`OTDRBackwardFileName`** must contain results of OTDR measurements performed in forward and backward directions of the same fiber link, so that the first event of the **`OTDRForwardFile`** is associated with the last event of the **`OTDRBackwardFile`**.
- In the case of multiple OTDR files used for simulation, the location of corresponding events must coincide with each other within the spatial resolution of the OTDR measurement for all specified OTDR files.

The OTDR trace at fixed frequency carries the information about:

- Attenuation of the span at this frequency $\alpha(f_i)$.
- EventLoss at event points $a_m(f_i)$.
- Event reflectance $R_m(f_i)$.
- Backscatter Coefficient $\beta_B(f_i)$ (optionally).
- Optical Return Loss $ORL(f_i)$ (optionally).

All these parameters may be used to find the backscatter coefficient of whole fiber links or for each fiber span. Each OTDR file contains the results of an OTDR measurement of the fiber at a fixed wavelength λ_{OTDR}. The Key Event block of an OTDR file contains important information about the following fiber characteristics: the number of events, event locations, event insertion loss, event reflectance and attenuation of the fiber spans. A typical Key event table is shown in Table 5.

Table 5 ***Typical Key Event table***

No.	Type	Location [km], Z_m	Reflect. [dB], R_m	Ins. Loss [dB], a_m	Atten. [dB/km], a	Cum.Loss [dB]	Info
1.	Reflect.	0.000	−31.62	0.00	-	-	
2.	Reflect.	25.146	−30.35	0.782	0.329	8.318	
3.	Non-Reflect.	37.864	-	0.391	0.338	13.344	
4.	Reflect.	50.520	−49.03	0.311	0.337	18.008	
5.	Non−Reflect.	63.238	-	−0.134	0.346	22.709	
6.	End	75.857	−13.61	-	0.344	26.914	

The behavior of the module when OTDR data is considered depends on many factors, but principally:

- the number of specified OTDR files for different λ_{OTDR}
- the directions at which OTDR files are measured (forward, backward, or both).

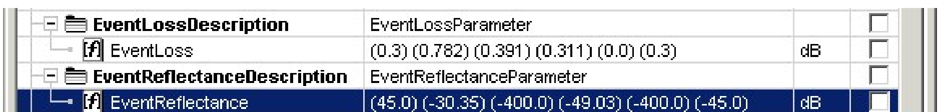
In the description below the term "single file" will be referred to the OTDR file which does not have a pair, measured at the same wavelength in opposite direction. The length of each fiber span is calculated as the difference between corresponding event locations $L_m = Z_{m+1} - Z_m$. If the data are provided by means of OTDR file(s), the parameter **`Length`** is ignored.

The Rayleigh backscatter coefficients are calculated using the backscatter coefficient (BSC) or optical return loss (ORL) data contained in the OTDR measurements. Thus at least one of these parameters must be supplied by the OTDR measurements. If no values for these parameters are provided by OTDR, then default values of BSC and ORL are presented in the OTDR file. These defaults will not be used for simulations — instead, the module uses the values specified by the parameter **`RayleighCoefficient`** or **`RayleighFilename`**. If both of the parameters BSC and ORL are supplied in the OTDR data, the module uses BSC. For the calculation of Rayleigh backscatter coefficient to be performed using ORL, the parameter BSC in the OTDR file must be set to the default value (80.0 dB).

The module behaves as follows.

If a single OTDR file is specified either by the parameters **`OTDRForwardFilename`** or **`OTDRBackwardFileName`**, the module uses OTDR data from the Key Event block (see columns 3–6 in Table 5) instead of the corresponding values specified by the parameters **`FiberLength`**, **`EventReflectance`**, **`EventLoss`**, **`Attenuation`**. The reflectance of the nonreflective events is assumed to be negligibly small (−400 dB). In the case of a single OTDR file, the value of the Rayleigh backscatter coefficients will be the same frequency-independent constant for all spans in the link.

Note: The reflectances of the fiber ends depend strongly on termination conditions which can differ between OTDR measurements and in the transmission system under simulation. Consequently, in practice these characteristics are not provided by OTDR measurements and, therefore, the event reflectance of the first and the last events must to be specified using one of the parameters **`EventReflectance`** or **`EventReflectanceFilename`**.
The insertion loss of the fiber ends (first and last events) are not supplied in the OTDR data, and the values of these parameters therefore are used as defined either by the parameter **`EventLoss`** or corresponding data file(s) specified by parameter **`EventLossFilename`**. For example, if an OTDR file containing data as specified in Table 5 to set the insertion loss and reflectance of both fiber ends to be equal 0.3 dB and −45 dB respectively, the parameters **`EventLoss`** and **`EventReflectance`** may be set to be arrays of six components, the first and the last of which should be equal to 0.3 for **`EventLoss`** and −45.0 for **`EventReflectance`**. The other array components (from the second to the fifth) will be ignored because in this case OTDR data is used.



Alternatively, to specify reflectance and loss identically for both fiber ends it is sufficient to specify the parameters

EventLoss and **EventReflectance** using a single value each.

Note: In unidirectional OTDR measurements, the standard procedure of processing OTDR traces can lead to unavoidable errors in calculation of insertion loss. In fact, the actual value of the event loss a_{m+1}^{act} differs from the result of OTDR measurement a_{m+1}^{meas}, according to the following equation [19].

$$a_{m+1}^{meas} = \alpha_{act}^{m+1} + 5log\frac{\eta_m}{\eta_{m+1}} , \tag{54}$$

Note: Where η_{m+1} and η_m are the fiber Rayleigh backscatter coefficients after and before the event located at the point $z = Z_{m+1}$. If η_{m+1} is higher than η_m, the value of the event loss obtained from OTDR measurement at this point will be underestimated and can even lead to negative values (see, for instance, line 5 of the Key event table in Table 5). Hence, in general, the measured event loss value depends on the direction of the OTDR measurement. To avoid such an ambiguity the results of OTDR measurements in both the forward and backward directions with the same wavelength should be used.

If a pair of OTDR files containing results of OTDR measurements with the same wavelength λ_{OTDR} in both the forward and backward directions is specified by the parameters **OTDRForwardFilename** and **OTDRBackwardFilename** respectively, the actual values of the event loss are calculated as a simple average of corresponding measurements in opposite directions. To improve accuracy, the averaging is applied to attenuation, event reflectance and event locations measured in both directions as well. For example, the averaged attenuation is defined by:

$$\alpha_{OTDR} = \frac{\alpha_{OTDR}^{AB} + \alpha_{OTDR}^{BA}}{2} . \tag{55}$$

Using (54) the ratio of Rayleigh backscatter coefficients of the fiber spans spliced at the event point are calculated as

$$\sigma_{m+1} = \frac{\eta_{m+1}}{\eta_m} = 10^{-\frac{[a_{m+1}^{AB} - a_{m+1}^{BA}]}{10}} , m = 1,2, \dots , K-2, \tag{56}$$

where a_{m+1}^{AB} and a_{m+1}^{BA} are the values of the event loss measured by OTDR in forward and backward directions correspondingly. The Rayleigh backscatter coefficients for each fiber span are evaluated from (56) with η_1 calculated using either backscatter coefficient or optical return loss supplied by the OTDR file.

The frequency dependence of attenuation, Rayleigh backscatter coefficient, event loss, and event reflectance may be accounted for by supplying multiple OTDR files with different frequency of OTDR measurements. A linear interpolation is applied to the data in the files to calculate the values of corresponding parameters between the specified frequency (wavelength) points. It is highly recommended to use the results of OTDR measurements in both forward and backward directions as discussed above. If several OTDR files are specified by using the parameters **OTDRForwardFilename** and **OTDRBackwardFilename**, then the frequency dependence of OTDR data is calculated using the data defined by parameter **OTDRForwardFilename**. In this case, OTDR data specified by parameter **OTDRBackwardFilename** will be used only if OTDR data with the same wavelength of measurements in forward directions is supplied.

If OTDR data is specified for a single frequency, the frequency dependence can be extracted from data file(s) as specified by the parameters **AttFileName**, **RayleighFilename**, **EventLossFilename**, and **EventReflectanceFilename**. In this case the values of OTDR data for other frequencies are calculated using the frequency dependence of corresponding physical parameters defined in the above mentioned data files. For example, the frequency dependence of the attenuation coefficient is calculated as follows:

$$\alpha(f_i) = \alpha^{DF}(f_i) + [\alpha^{OTDR}(f_{OTDR}) - \alpha^{DF}(f_{OTDR})] , \tag{57}$$

where $\alpha^{DF}(f_i)$ is the frequency dependence specified by the user data file **AttFileName**, and $\alpha^{OTDR}(f_{OTDR})$ is given by OTDR data. It means that attenuation specified in the file shifted up or down in order the obtained curve to be passed through the points specified in OTDR file (see Figure 14). If a negative value of the attenuation coefficient is obtained in such a calculation it is replaced by zero. A similar procedure is applied to calculate frequency dependence of the Rayleigh backscatter coefficient, event loss, and event reflectance.

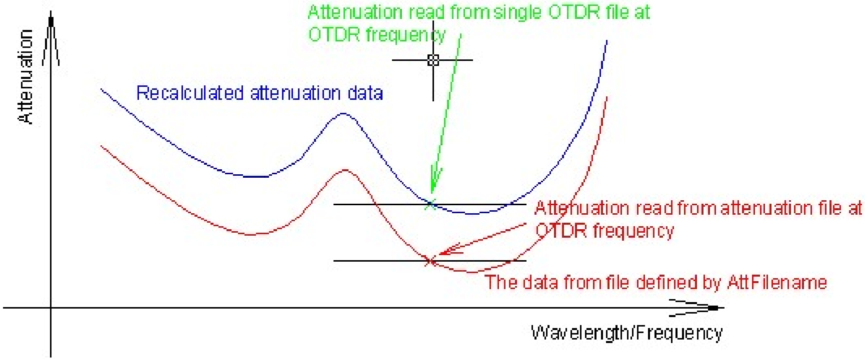


Figure 14 Schematic representation of attenuation recalculation when attenuation file and single frequency OTDR file (s) are specified

OTDR Debug File Description

OTDR data such as attenuation, Rayleigh backscatter coefficient, losses and reflectances at event points may be stored in a file defined by the enhanced parameter **`OTDRDebugFilename`**. If a pair of files (forward + backward) at the same frequency is provided, the mean averaged values for event loss, event reflectance and attenuation are stored. A sample file is shown below:

```
#OTDR Frequency: 193.414 THz
#Direction: Forward&Backward
Forward BackScatteringCoeff: 81.5 dB
Backward BackScatteringCoeff: 81.5 dB
Forward ORL: 0 dB
Backward ORL: 0 dB
Span No. 1
Span Length: 725.817 m
Attenuation: 2.030000e-004 dB/m
RayleighCoeff: 3.473933e-008 1/m
Event No. 2
EventLoss: 0.1975 dB
EventReflectance: -400 dB
Span No. 2
Span Length: 1484.12 m
Attenuation: 1.990000e-004 dB/m
RayleighCoeff: 3.485953e-008 1/m
```

File Formats

Many of the input files accepted by the *UniversalFiber* describe one or more dependent fiber properties as a function of an independent physical quantity such as frequency of wavelength. All files of this type follow a standard format, which is described below. Other input files — such as **`RamanFilename`**, for example — have formats that are better suited to their specialized functions. Note that **`RamanFilename`** files may have a different form for scalar and vectorial cases. The formats of these files are also described below.

Definitions

The symbols shown as underlined (and in red) are obligatory words in headers and separators, such as, "[", or "]", or "#", or "(", or ")", <quantity> — the values of specified quantity, <unit> — the unit of specified quantity. File specifications and examples are described below.

Each table contains the mask of the file and the examples.

Files with Simple Format

Table 6 ***Format of AttenuationFilename and examples***

# Frequency/Wavelength Attenuation	# Wavelength Attenuation	# Frequency Attenuation
# (<FrequencyUnit/WavelengthUnit>) (<AttenuationUnit>)	# (nm) (dB km^-1)	# (THz) (dB m^-1)
<Frequency_1/Wavelength_1> <Attenuation_1>	850 1.81	352.7 1.81e-3
...	1300 0.35	230.6 0.35e-3
<Frequency_n/Wavelength_n> <Attenuation_n>	1550 0.19	193.4 0.19e-3

Table 7 ***Format of DispersionFilename and examples***

# Frequency/Wavelength Dispersion	# Wavelength Dispersion	# Frequency Dispersion
# (<FrequencyUnit/WavelengthUnit>) (<DispersionUnit>)	# (nm) (ps nm^-1 km^-1)	# (THz) (s m^-2)
<Frequency_1/Wavelength_1> <Dispersion_1>	1500 14.34	199.9 14.34e-6
...	1550 17.38	1.934 17.38e-6
<Frequency_n/Wavelength_n> <Dispersion_n>	1600 20.19	187.4 20.19e-6

Table 8 ***Format of RayleighFilename and examples***

# Frequency/Wavelength RayleighCoefficient	# Wavelength RayleighCoefficient	# Wavelength RayleighCoefficient
# (<FrequencyUnit/WavelengthUnit>) (<RayleighCoeffUnit>)	# (nm) (dB)	# (nm) (neper m^-1)
<Frequency_1/Wavelength_1> <RayleighCoeff_1>	1310 -77	1310 9.79e-08
...	1550 -82	1550 3.09e-08
<Frequency_n/Wavelength_n> <RayleighCoeff_n>		

Table 9 ***Format of SBSGainFilename and examples***

# Frequency/Wavelength SBSGain		# Wavelength SBSGain	# Frequency SBSGain
# (<FrequencyUnit/WavelengthUnit>) (<SBSGainUnit>)		# (nm) (m W^-1)	# (THz) (m W^-1)
<Frequency_1/Wavelength_1> <SBSGain_1>		1500 4.5e-11	199.9 4.5e-11
...		1550 4.6e-11	1.934 4.6e-11
<Frequency_n/Wavelength_n> <SBSGain_n>		1600 4.7e-11	187.4 4.7e-11

Table 10 ***Format of CoreAreaFilename and examples***

# Frequency/Wavelength CoreArea	# Wavelength CoreArea	# Frequency CoreArea
# (<FrequencyUnit/WavelengthUnit>) (<CoreAreaUnit>)	# (nm) (um^2)	# (THz) (m^2)
<Frequency_1/Wavelength_1> <CoreArea_1>	1310 66.5	230.6 66.5e-12
...	1550 84.9	193.1 84.9e-12
<Frequency_n/Wavelength_n> <CoreArea_n>		

Table 11 ***Format of EventLossFilename and examples***

# Frequency/Wavelength EventLoss		# Wavelength EventLoss	# frequency EventLoss
# (<FrequencyUnit/WavelengthUnit>) (<EventLossUnit>)		# (nm) (dB)	# (Hz) (dB)
<Frequency_1/Wavelength_1> <EventLoss_1>		1500 3.1	199.9e12 3.1
...		1550 3.3	193.4e12 3.3
<Frequency_n/Wavelength_n> <EventLoss_n>		1560 3.7	187.4e12 3.7

The second variant:

# Frequency/Wavelength TransmissionCoefficient		# Wavelength TransmissionCoefficient	# frequency TransmissionCoefficient
# (<FrequencyUnit/WavelengthUnit>) (<TransmissionCoefficientUnit>)		# (m) (.)	# (Hz) (.)
<Frequency_1/Wavelength_1> <TransmissionCoefficient_1>		1400e-9 0.21	2.141E+14 0.21
...		1550e-9 0.23	1.934E+14 0.23
<Frequency_n/Wavelength_n> <TransmissionCoefficient_n>		1600e-3 0.27	1.874E+14 0.27

Table 12 ***Format of EventReflectanceFilename***

# Frequency/Wavelength EventReflectance		# Wavelength EventReflectance	# frequency EventReflectance
# (<FrequencyUnit/WavelengthUnit>) (<EventReflectanceUnit>)		# (nm) (dB)	# (THz) (dB)
<Frequency_1/Wavelength_1> <EventReflectance_1>		1480 -55	199.9 -56
...		1550 -61	193.4 -61
<Frequency_n/Wavelength_n> <EventReflectance_n>		1560 -62	187.4 -63

The second variant:

# Frequency/Wavelength ReflectionCoefficient		# Wavelength ReflectionCoefficient	# frequency ReflectionCoefficient
# (<FrequencyUnit/WavelengthUnit>) (<ReflectionCoefficientUnit>)		# (nm) (dB)	# (THz) (dB)
<Frequency_1/Wavelength_1> <ReflectionCoefficient_1>		1480 3.16E-06	199.9 3.16E-06
...		1550 7.94E-07	193.4 7.94E-07
<Frequency_n/Wavelength_n> <ReflectionCoefficient_n>		1560 6.31E-07	187.4 5.01E-07

Table 13 ***Format of NonlinearFilename***

# Frequency/Wavelength NonlinearCoefficient		# Wavelength NonlinearCoefficient
# (<FrequencyUnit/WavelengthUnit>) (<NonlinearCoefficientUnit>)		# (nm) (m^-1 W^-1)
<Frequency_1/Wavelength_1> <NonlinearCoefficient_1>		1310 1.2e-4
...		1550 2.5e-4
<Frequency_n/Wavelength_n> <NonlinearCoefficient_n>		1560 2.8e-4

Format of RamanFilename

For scalar cases, that is, cases where the parameter **PolarizationAnalysis** = Scalar, if the file contains a real and an imaginary parts, the Raman profiles have the following format:

```
# df/dlambda Raman_Im Raman_Re
# (dfUnit/dlambdaUnit) (<Raman_ImUnit>) (<Raman_ReUnit>)
# PumpFrequency1/PumpWavelength1 [<PumpFrequency1Unit/PumpWavelength1Unit>]
<PumpFrequency1/PumpWavelength1>
<df_1/dlambda_1> <Raman_Im_1> <Raman_Re_1>
...
<df_n/dlambda_n> <Raman_Im_n> <Raman_Re_n>
# PumpFrequency2/PumpWavelength2 [<PumpFrequency2Unit/PumpWavelength2Unit>]
<PumpFrequency2/PumpWavelength2>
<df_1/dlambda_1> <Raman_Im_1> <Raman_Re_1>
...
<df_n/dlambda_n> <Raman_Im_n> <Raman_Re_n>
...
```

The file may contain one or a number of blocks corresponding to certain pump frequencies or wavelengths.

If a Raman gain profile only is specified, the file will have the following format:

```
# df/dlambda Raman_Im/RamanGain
# (dfUnit/dlambdaUnit) (<Raman_ImUnit/RamanGainUnit>)
# PumpFrequency1/PumpWavelength1 [<PumpFrequency1Unit/PumpWavelength1Unit>]
<PumpFrequency1/PumpWavelength1>
<df_1/dlambda_1> <Raman_Im_1/RamanGain_1>
...
<df_n/dlambda_n> <Raman_Im_n/RamanGain_n>
...
```

Note that the *df* and *dlambda* should be positive values, which means that the *df* (specified of calculated using *dlambda*) is subtracted from the pump frequency.

Vector case, that is, the parameter **PolarizationAnalysis** = VectorPMD:

These types of file defined by **RamanFilename** may be applied to vectorial cases only: in this case, copolarized and transversal gains are specified. The file has the following format:

```
# df/dlambda CodirectionalRaman_Im CodirectionalRaman_Re TransversalRaman_Im
TransversalRaman_Re
# (dfUnit/dlambdaUnit) (<CodirRaman_ImUnit>) (<CodirRaman_ReUnit>) (<TransvRaman_ImUnit>)
(<TransvRaman_ReUnit>)
# PumpFrequency1/PumpWavelength1 [<PumpFrequency1Unit/PumpWavelength1Unit>]
<PumpFrequency1/PumpWavelength1>
<df_1/dlambda_1> <CodirRaman_Im_1> <CodirRaman_Re_1> <TransvRaman_Im_1>
<TransvRaman_Re_1>
...
```

In most practical cases, transversal Raman gain is assumed to be negligible. The last two columns may not be specified.

```
# df/dlambda CodirectionalRaman_Im CodirectionalRaman_Re
# (dfUnit/dlambdaUnit) (<CodirRaman_ImUnit>) (<CodirRaman_ReUnit>)
# PumpFrequency1/PumpWavelength1 [<PumpFrequency1Unit/PumpWavelength1Unit>]
<PumpFrequency1/PumpWavelength1>
<df_1/dlambda_1> <CodirRaman_Im_1> <CodirRaman_Re_1>
...
```

If the real part of copolarized gain is not provided, we assume that Raman response is instantaneous.
In this case, the file may contain only one column:

```
# df/dlambda CodirectionalRaman_Im
# (dfUnit/dlambdaUnit) (<CodirRaman_ImUnit>)
# PumpFrequency1/PumpWavelength1 [<PumpFrequency1Unit/PumpWavelength1Unit>]
<PumpFrequency1/PumpWavelength1>
<df_1/dlambda_1> <CodirRaman_Im_1>
...
```

Note: The file format for scalar cases may also be used for vector ones. The second and the third columns (if provided) are interpreted as an imaginary part and a real part of copolarized Raman gain.

Table 14 ***Examples of Raman files***

```
# df RamanGain
# (THz) (W^-1 m^-1)
# PumpFrequency [THz] 205.337
// 1460 nm pump
0.00000E+00 0.00000E+00
2.83665E-02 1.05990E-06
1.36612E+01 6.57743E-04
...
2.96647E+01 7.78414E-05

# PumpFrequency [THz] 199.862
// 1500 nm pump
0.00000E+00 0.00000E+00
7.15274E-02 2.41228E-06
...
```

```
# df Raman_Im Raman_Re
# (Hz) (W^-1 m^-1) (rad W^-1
m^-1)
# PumpWavelength [nm] 1550
0.00E+00 0.00E+00 2.63E-04
5.00E+10 1.34E-06 2.63E-04
5.80E+12 2.02E-04 2.82E-04
1.14E+13 6.76E-04 1.98E-04
...
```

Table 15 ***Format of OverlapIntegralFilename***

```
# Frequency2/Wavelength2 [<Frequency2Units/Wavelength2Units>] OverlapIntegral
[<OverlapIntegralUnits>]
# Frequency1/Wavelength1 [<Frequency1Units/Wavelength1Units>]
<Frequency1/Wavelength1>
<Frequency2_1/Wavelength2_1> <OverlapIntegral_1>
...
<Frequency2_n/Wavelength2_n> <OverlapIntegral_n>
# Frequency1/Wavelength1 [<Frequency1Units/Wavelength1Units>]
<Frequency1/Wavelength1>
<Frequency2_1/Wavelength2_1> <OverlapIntegral_1>
...
<Frequency2_n/Wavelength2_n> <OverlapIntegral_n>
```

What follows is an example of the file that contains overlap integral data:

```
# Wavelength2 [nm] OverlapIntegral
[m^-2]
# Wavelength1 [nm] 1550
1510 1.51E+10
```

```
1530 1.49E+10
1550 1.46E+10
1570 1.43E+10
1590 1.41E+10
1610 1.39E+10
# Wavelength1 [nm] 1570
1510 1.49E+10
1530 1.46E+10
1550 1.43E+10
1570 1.41E+10
1590 1.39E+10
1610 1.36E+10
1630 1.34E+10
```

Table 16 contains the parameters that can be specified in input files.

Table 16 ***Parameters that can be specified in files***

Parameter	**Parameter name**	**Keyword**	**Units**	**Units specified in the file**
Frequency	**In all the files except RamanFilename**	Frequency	Hz GHz THz	(Hz) (GHz) (THz)
Wavelength	**In all the files except RamanFilename**	Wavelength	m um nm	(m) (um) (nm)
Pump frequency	**RamanFilename**	PumpFrequency	Hz GHz THz	[Hz] [GHz] [THz]
Frequency shift	**RamanFilename**	df	Hz GHz THz	(Hz) (GHz) (THz)
Wavelength shift	**RamanFilename**	dlambda	m um nm	[m] [um] [nm]
Attenuation	**AttFilename**	Attenuation	dB/m dB/km 1/m 1/km	(neper m^-1) (neper km^1) (dB m^-1) (dB km^-1)
Dispersion	**DispersionFilename**	Dispersion	s/m^2 ps/nm/km	(s m^-2) (ps nm^-1 km^-1)
Core Area	**CoreAreaFilename**	CoreArea	m^2 mm^2 um^2	(m^2) (mm^2) (um^2)
OverlapIntegral	**OverlapIntegralFilename**	OverlapIntegral	m^{-2} mm^{-2} um^{-2}	(m^-2) (mm^-2) (um^-2)
NonlinearCoefficient	**NonLinearFilename**	NonlinearCoefficient	1/m/W	(m^-1 W^-1)
RayleighCoefficient	**RayleighFilename**	RayleighCoefficient	dB/m dB/km 1/m 1/km or dB	(neper m^-1) (neper km^1) (dB m^-1) (dB km^-1)
SBS Gain	**SBSGainFilename**	SBSGain	m/W	(m W^-1)
Event Loss	**EventLossFilename**	EventLoss	dB	(dB)
Transmission Coefficient	**EventLossFilename**	TransmissionCoefficient		(.)
Event Reflectance	**EventReflectanceFilename**	EventReflectance	dB	(dB)
Reflection Coefficient	**EventReflectanceFilename**	ReflectionCoefficient		(.)

Raman gain (Codirectional, Transversal)	**RamanFilename**	RamanGain RamanIm CodirectionalRaman_Im TransversalRaman_Im	1/m/W 1/km/W	(1/m/W) (m^-1 W^-1) (W^-1 m^-1) (1/km/W) (km^-1 W^-1) (W^-1 km^-1)
Raman phase response	**RamanFilename**	RamanRe CodirectionalRaman_Re TransversalRaman_Re	1/m/W 1/km/W rad/m/W rad/km/W	(1/m/W) (m^-1 W^-1) (W^-1 m^-1) (1/km/W) (km^-1 W^-1) (W^-1 km^-1) (rad/m/W) (rad m^-1 W^-1) (rad W^-1 m^-1) (rad/km/W) (rad km^-1 W^-1) (rad W^-1 km^-1)

Acknowledgments

[1] N. R. Newbury, National Institute of Standards and Technology.