Active-FDTD

for OmniSim & CrystalWave

Version 5.0

Photon Design®

propagating ideas

34 Leopold St
Oxford, OX4 1TW
United Kingdom
Phone + 44 1865 324990
Fax + 44 1865 324991
info@photond.com
www.photond.com

Photon Design
34 Leopold St
Oxford, OX4 1TW
United Kingdom

Phone +44 1865 324 990
Fax +44 1865 324 991
Email: support@photond.com

CONTENTS

Chapter 1 Introduction

1.1 Static Gain and Dynamic Gain models

The Active-FDTD Module for CrystalWave and OmniSim provides the ability to include gain in an FDTD simulation for the modelling of active devices including semiconductor lasers. The module can operate in two different modes:

a) Static Gain

In Static Gain mode, the model does not explicitly consider a carrier density in the device but rather uses a saturable gain:

$$G(P) = \frac{G_0}{1 + \varepsilon_{sat} P}$$

where P is the photon density, ε_{sat} is the gain non-linearity constant. Although this is strictly speaking not a true representation of how carriers interact with an optical beam, it is often a sufficient approximation.

b) Dynamic Gain

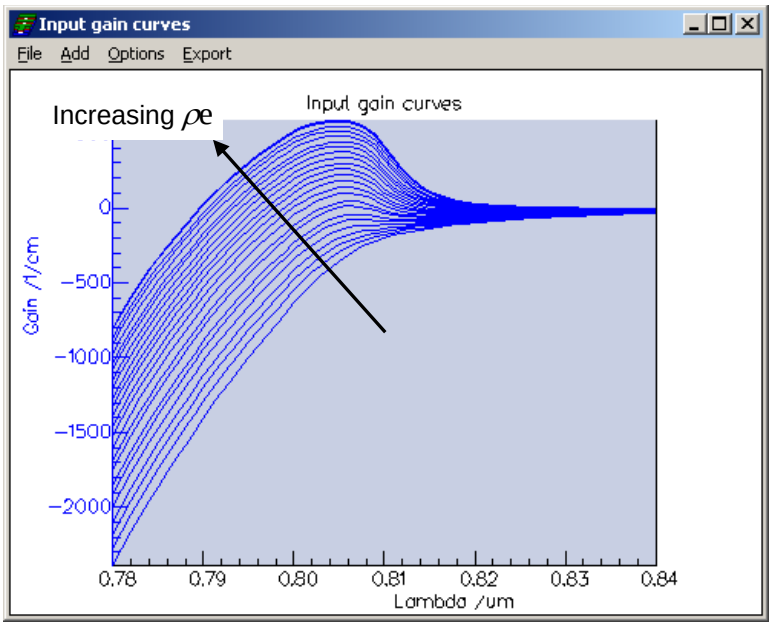
In this case the model includes a rate equation for a carrier density ρ in the active layer of the form:

$$\frac{d\rho}{dt} = -S - \frac{\rho}{\tau(\rho)} + \frac{J}{ed}$$

where $\tau(\rho)$ is the carrier lifetime, J is the current injection, d is the active layer thickness and S is the stimulated recombination rate per unit volume, i.e. the carriers lost to stimulated recombination (i.e. gain) processes

1.2 Dynamic Gain model: gain curves

The Active-FDTD Dynamic Gain model can take a set of gain curves, either measured or simulated. The gain curves define the material gain as a function of carrier density and wavelength. You will need to supply sets of gain spectra for the whole range of carrier densities that might occur in your simulation. A typical set of gain curves might look as follows.



Above: a typical set of gain curves for a GaAs/AlGaAs active material showing gain spectra for increasing carrier density.

The Active-FDTD engine models the gain spectrum at a particular carrier density as a superposition of 4 Lorentzians, with each Lorentzian being characterised by a peak position (in radians/s), the inverse of the half angular width (in s) and amplitude (gain peak in 1/cm). Thus it characterises each spectrum by 12 parameters. Of course each parameter is carrier-dependent. The engine uses a polynomial to define this carrier dependence of each of the 12 parameters. These 12 polynomials are defined via the Material Database. The Module provides a fitting tool to generate the 12 polynomials from a set of gain spectra. In principle however you could generate these polynomials from any other source you wish.

Chapter 2

Setting up an active device

There are a variety of parameters in different places that are required in order to simulate an active device. We list all the parameters here.

2.1 Material Database: Gain Definition

The frequency dependent material gain is modelled using a given number of Lorentzians. It is given by

$$G(\omega) = \frac{1}{1+\varepsilon_{sat} P} \sum_{k=1}^{M} L_k(\omega)$$

where ε_{sat} is the gain saturation constant (in $1/\text{cm}^3$), P the photon density (in cm^3). The formula for the k-th Lorentzian is

$$L_k(\omega) = \frac{G_{peak,k}}{2}\left(\frac{1}{1+j(\omega-\omega_k)T_k} + \frac{1}{1+j(\omega+\omega_k)T_k}\right)$$

where $G_{peak,k}$ is the gain peak (in $1/\text{cm}$), T_k (in s) is the inverse of half of the angular frequency width $\Delta\omega_k$

$$T_k = 2/\Delta\omega_k$$

and ω_k is the angular frequency (in radians/s) of the maximum gain.

As explained in the introduction, the carrier-dependent gain spectra of the Dynamic Gain model are input via 12 polynomials with carrier density (in $1/\text{cm}^3$) as the variable, representing the carrier dependence of $G_{peak,k}$, T_k and ω_k of the 4 Lorentzians.

Thus for example, the polynomial for the gain peak of the k-th Lorentzian would look like

$$G_{peak,k} = g_{0k} + g_{1k}\rho + g_{2k}\rho^2 + g_{3k}\rho^3 + g_{4k}\rho^4 + \ldots + g_{polyOrder,k}\rho^{polyOrder}$$

where g_{ik} are the polynomial coefficients, ρ is the carrier density (in $1/\text{cm}^3$) and similarly for the other Lorentzian parameters. These polynomials are entered in the definition of the material in the Material Database. The syntax is:

Table 1. Material Parameters – gain curves for Dynamic Gain model				
Flag	*param1*	*param2*	*param3*	*param4*
GAIN_4LORENTZPOLY	polyOrder	confinementFactor (applies only for 2D)	minCarrierDensity [1/cm3]	maxCarrierDensity [1/cm3]

	{12 lines follow, each specifying the coefficients of one polynomial for one Lorentzian parameter; lines are in the following order: $G_{peak,1}$, … $G_{peak,4}$, T_1, … T_4, ω_1 … ω_4, where 1 to 4 represent the 4 Lorentzians. Each line is of the form c_0, c_1… c_polyOrder specifying the coefficients of the polynomial}
	Defines the 12 polynomials for the 4-Lorentzian gain function. **PolyOrder** is the order of the polynomials and the min/max parameters define the range of carrier density over which the polynomials are valid. The **confinementFactor** parameter is the *confinement factor* of the active layer – used for 2D FDTD simulations, defined as Γ in G_mode = Γ.G_mat, where G_mode is the mode gain, and G_mat = G_qw is the material gain of (presumably) the quantum wells (averaged over the wells if you have more than one well), given by the 4-Lorentzian function defined here. Note that you should enter just the vertical part of the confinement factor – if your laser mode extends laterally beyond the gain medium then this is taken into account implicitly in the simulation (3D simulations do not need a confinement factor at all since this overlap of the mode with the gain is explicitly computed in the simulation.).

In the case of the Static Gain model, the gain function is defined as a sum of Lorentzian functions, whose amplitude $G_{peak,k}$ (1/cm), position λ_k (nm) and width $\Delta\lambda_k$ (nm) need to be defined. The formula for each Lorentzian is the same as above. To convert these into the parameters of the Lorentzian, the application uses

$$\omega_k = \frac{2\pi c}{\lambda_k}$$

and the width $\Delta\lambda_k$ is related to T_k approximately by

$$T_k \approx \frac{4\pi c}{\omega_k^2 \, \Delta\lambda_k}$$

for $\omega_k >> \Delta\omega_k$. The syntax is:

Table 2. Material Parameters – gain spectrum for Static Gain Model				
Flag	*param1*	*param2*	*param3*	*param4*
GAIN_STATICLORENTZ	N	confinementFactor (applies only for 2D)		
	{N defines the number of lines that follow, each specifying the parameters of one of the Lorentzians used to define the spectrum}			
	Defines the gain function, sum of a number of Lorentzians specified by the **N**. The **confinementFactor** parameter is defined as above. On each line you should specify the amplitude of the Lorentzian gain function with peakGain (in 1/cm), its position with peakPosn (in nm) and its spectral width (full width at half maximum) using peakWidth (nm).			

An example is shown below, combining both dynamic and static gain definition.

```
GAIN_4LORENTZPOLY   5   0.05   1.426900e+018   2.922500e+018   //   polyOrder   confFactor
minCarrierDensity maxCarrierDensity
-8.830578e+004  1.657169e-013 -1.396054e-031  5.833995e-050 -1.150742e-068  8.269638e-088
 4.999274e+004 -1.134082e-013  1.033017e-031 -4.734815e-050  1.101760e-068 -1.037551e-087
 5.707949e+004 -1.434475e-013  1.380566e-031 -6.474787e-050  1.486592e-068 -1.340330e-087
-8.723360e+004  2.088095e-013 -1.959599e-031  9.051644e-050 -2.065428e-068  1.864994e-087
-2.425854e-012  5.725397e-030 -5.223742e-048  2.317563e-066 -4.965831e-085  4.103904e-104
-4.758226e-012  1.051797e-029 -9.055686e-048  3.842308e-066 -8.067461e-085  6.722902e-104
 2.552647e-012 -6.356568e-030  6.249060e-048 -3.000677e-066  7.068528e-085 -6.547759e-104
 3.437163e-011 -8.329089e-029  7.914640e-047 -3.686681e-065  8.430648e-084 -7.584424e-103
 2.890517e+015 -9.483570e-004  8.333869e-022 -3.654716e-040  7.743815e-059 -6.250535e-078
 3.391884e+014  4.417437e-003 -3.873510e-021  1.686473e-039 -3.645113e-058  3.128928e-077
 1.236435e+015  2.600814e-003 -2.317522e-021  1.021371e-039 -2.215876e-058  1.890052e-077
 1.450466e+016 -2.926253e-002  2.731156e-020 -1.245109e-038  2.782714e-057 -2.446104e-076
//-------------------------------------------------------------------------------------
GAIN_STATICLORENTZ 2 0.1 // N confFactor
 1300 851 42  // peakGain [1/cm] peakPosn [nm] peakWidth [nm]
 300  859 29  // peakGain [1/cm] peakPosn [nm] peakWidth [nm]
```

2.2 Material Database: Other Material Parameters

You must define a number of other parameters in the material database in order to use a gain material. The table below lists these, and additional details are given further on.

Table 3. Material Parameters – spontaneous lifetime and gain non-linearity					
Flag	***Units***	***param1***	***param2***	***param3***	***param4***
RADRECOMB	[1/s]	tr0 (ps)	Br [cm^3/s]	Cr [cm^6/s]	
	Defines radiative recombination rate: $d\rho/dt = -\rho/tr_0 - Br\cdot\rho^2 - Cr\cdot\rho^3$ (But see notes below)				
NRADRECOMB	[1/s]	tn0 (ps)	Bn [cm^3/s]	Cn [cm^6/s]	
	Defines non-radiative recombination rate: $d\rho/dt = -\rho/tn_0 - Bn\cdot\rho^2 - Cn\cdot\rho^3$ (But see notes below)				
GAIN_EPS	[cm^3]	ε_{sat} [cm^3]			
	Gain saturation parameter, reduces G_{peak} as follows: $G'_{peak} = G_{peak}/(1 + \varepsilon_{sat}.P)$, where P is the local photon density in $1/cm^3$ where G_{peak} is given by GAIN_4LORENTZPOLY or GAIN_STATICLORENTZ.				

2.2.1 Refractive Index

In your active material you must also define the refractive index via the material database, e.g. via the RIX_POLYX flag – see main application documentation. Note that the gain materials use a refractive index which is constant with wavelength. So even if you define the wavelength dependence in the material database, the value at the *central wavelength* of the FDTD simulation will be used for all wavelengths. Thus all the parameters to do with refractive index fitting in the *FDTD Material Parameters* panel are ignored for gain materials.

2.2.2 Spontaneous Lifetime

You must define the spontaneous recombination lifetime of the carriers in your active medium. This is done via extensions to the material - the RADRECOMB and NRADRECOMB flags as shown in Table 2. The radiative recombination is associated with the generation of spontaneous photons; in contrast non-radiative recombination does not generate photons.

Note 1 This version of the Active-FDTD model does not model spontaneous emission – i.e. no photons are injected into the FDTD simulation by the radiative recombination process. See also §5.1.

Note 2 This version ignores the B and C terms of these recombination flags – i.e. the recombination rate (or lifetime) is assumed to be constant with carrier density. The additional parameters are present for compatibility with other active products of Photon Design and for future enhancement.

2.2.3 Gain Non-linearity

You must also define the gain non-linearity for your active material – see GAIN_EPS above. You can turn off non-linear gain effects by setting a zero value for GAIN_EPS.

2.3 Defining the active region

The choice of the active region is defined differently for 2D and 3D FDTD calculations.

2.3.1 Selecting the material database

The material database for the device must be set to the material database in which the gain material is defined.

The material database is set via the **MaterialsDatabase** parameter in the *Device Properties* panel.

2.3.2 2D FDTD: Mask Layer Properties

In a 2D Active FDTD calculation, active areas are associated with an active Mask Layer. The properties of the Mask Layers are located in the *Mask Layer Properties for 2D Calculations* panel, which are accessed from the *Object Layer Palette for 2D Calculations*.

The following parameters need to be set:

Mask2DSpec Must be set to *Mask2DMaterial*.

Mask2DMaterial Must be set to the relevant gain material selected from the material database.

IsActive If true, then this area of your 2D-FDTD calculation will be active

TrueActiveThickness Not used in 2D

2.3.3 3D FDTD: Physical Layer Properties

In a 3D Active FDTD calculation, active areas are associated with an active Physical Layer. The properties of the Physical Layers are located in the *Physical Layer Properties* panel, which are accessed from the *Physical Layer Palette*.

The following parameters need to be set:

Spec Must be set to *Material*.

Material Must be set to the relevant gain material selected from the material database.

IsActive If true, then this layer of your 3D-FDTD calculation will be active, provided it is not etched away.

TrueActiveThickness [um] A quantum well laser typically has several very thin quantum wells separated by barriers. Rather than entering all the quantum wells in your structure explicitly, you can create one averaged layer corresponding to all the quantum wells and barrier layers. However remember that the gain spectrum is always defined in the material database file as material gain (with mode gain = (confinement factor)*(material gain)). Thus the program needs to know how much of your averaged layer is quantum well, so that the waveguide mode gain is correct. This is the function of the **TrueActiveThickness** and should be set to the sum of the widths of all the quantum wells.

If you enter all the quantum wells explicitly, remember to set **TrueActiveThickness** to 1.

2.4 Electrical Sources

For a Dynamic Gain simulation, you need to inject current into your active layers. For this purpose you must add an *electrical source*. To add a source:

- Click on the toolbar to display the excitor and detector tools:

  The electrical source is the green square. Select this. Your cursor should change to show the electrical source tool.

- Move your cursor to the top-left corner of where you want your source to be. Press the left button down and drag the cursor to the bottom-right corner and then release the left button.
- If you want to define the exact position and size of the source you can right-click on the source and edit its position and size properties explicitly.

2.4.1 Vertical Extent of Electrical Source (3D FDTD only)

The electrical source can be set up to pump one or more active layers by defining the Top Boundary and Bottom Boundary properties. The electrical source will pump all active layers that it finds between its top and bottom boundaries, sharing its current between them.

If you have e.g. 3 active layers between the source's boundaries then the sources current density will be divided between the 3 active layers according to the **TrueActiveThickness** of each layer. For example layer 2 would get a current density J2=t2/(t1+t2+t3)*Jsource where t1, t2, t3 are the **TrueActiveThickness**'s of each layer.

Top Boundary		
YMaxSpec	how the y-coordinate of the top boundary of the excitor is specified: • **Top of device** • **Bottom of device** • **Top of layer** • **Centre of layer** • **Bottom of layer** • **Coordinate**	
YMaxLayer	if **YMaxSpec** is set to **Top/Centre/Bottom of layer** then the y-coordinate of the top boundary of the excitor is set to the top/centre/bottom of this *physical layer*	
Bottom Boundary		
YMinSpec	how the y-coordinate of the bottom boundary of the excitor is specified: • **Top of device** • **Bottom of device** • **Top of layer** • **Centre of layer** • **Bottom of layer** • **Coordinate**	
YMinLayer	if **YMinSpec** is set to **Top/Centre/Bottom of layer** then the y-coordinate of the bottom boundary of the excitor is set to the top/centre/bottom of this *physical layer*	

2.4.2 Time Evolution of Electrical Source

Via the electrical source properties editor, you can control how the source current varies in time. Mostly a dc current will suffice but it is possible to define a near arbitrary time-dependence.

Time Domain		
TimeEvolvingSignal	If set to *Constant* then the StartBias parameter is used to define a constant current. If set to *Modulated Ramp* then the following 5 parameters are used to define a flat or linearly changing current, with an optional sinusoidal modulation superimposed. If set to *Generic Function* then you can enter a general expression in the **GenericFunction** property	
StartBias	(used for *Constant* and *Modulated Ramp* type). Defines I1 – see below.	mA
EndBias	(used for *Modulated Ramp* type). Defines I2 – see below.	mA
ModulationAmplitude	(used for *Modulated Ramp* type). Defines A – see below.	mA
ModulationPeriod	(used for *Modulated Ramp* type). Defines T – see below.	fs
DurationPeriod	(used for *Modulated Ramp* type). Defines the duration of the signal. Once the simulation time is greater than **DurationPeriod**, the signal is re-set and starts again, creating a periodic signal.	fs
GenericFunction	(used for *Generic Function* type). Enter an expression for the source current as a function of time. Time is denoted by the parameter "t" and is in units of fs.	mA

2.5 FDTD Calculator parameters

Once you have set up all the parameters described above you must open the FDTD Calculator.

2.5.1 Carrier Solver parameters

In the FDTD Parameters panel you will find a Carrier Solver section which you must set up for dynamic gain simulations:

Carrier Solver		
CarrierAccelFactor	Typically the carrier lifetime is many orders of magnitude longer than what can be realistically simulated in an FDTD run. Therefore using this factor we can increase the speed of the carrier dynamics artificially using this factor. (See also 5.2.)	
CarrierSpatialResolution	It is not necessary to solve the carrier rate equations with a spatial resolution equal to that needed in solving Maxwell's Equations in the FDTD algorithm. The resolution used for the carrier density is therefore given by: **GridSpacing*CarrierSpatialResolution** where **GridSpacing** is the FDTD grid size.	
InitialCarrierDensity	The carrier density in the active layer(s) will be set to this value at the beginning of the FDTD calculation.	cm^{-3}

2.5.2 Material parameters

To use a gain material, you must enable it in the *FDTD Material Parameters* panel, which you can open by double-clicking on the field next to **Materials** in the *FDTD Calculator*.

For each gain material in your design, select the material in the list and set the following properties.

Model Set this to *Static Gain* to use the Static Gain model or *Dynamic Gain* to use the Dynamic Gain model.

GainMultiplier (Static Gain only) This allows you to increase the gain you have specified in the material database according to the factor you specify here. This parameter is solely for convenience, allowing you to adjust the gain conveniently through the graphical user interface.

Note:

If you set **Model** to *Static Gain* then you must set the GAIN_STATICLORENTZ flag for this material in the material database. If you set **Model** to *Dynamic Gain* then you must set the GAIN_4LORENTZPOLY flag in the material database – see also 2.1.

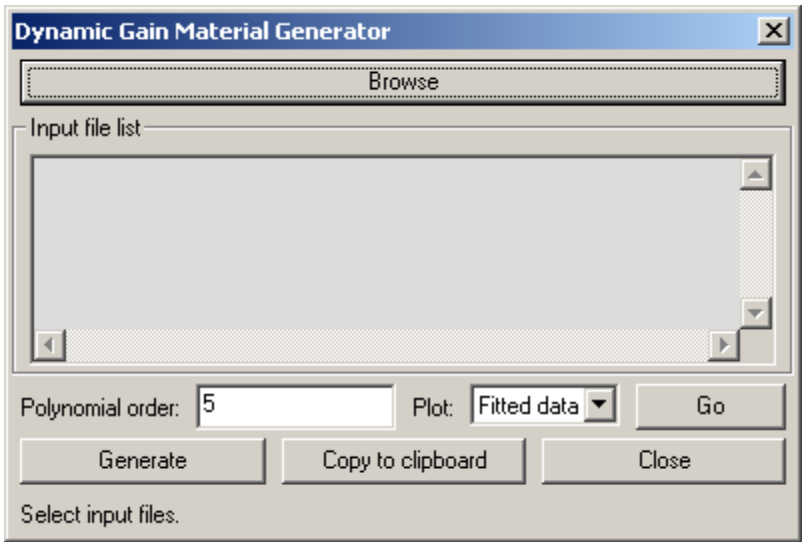
Chapter

3

The Dynamic Gain Material Generator

As explained previously, the Active-FDTD Module needs to know the material gain as a function of wavelength and carrier density. For this purpose it uses 12 parameters to define a gain spectrum at a given carrier density and defines the carrier-density dependence of these 12 parameters using polynomials.

The Dynamic Gain Material Generator is a tool that is able to read a set of gain spectra at a range of carrier densities and generate the 12 polynomials required for the Active-FDTD Module. The Generator is shown below.



Above: the Dynamic Gain Material Generator interface.

3.1 The Gain Spectra Files

The gain spectra are provided to the Generator in a set of files, one file per spectrum. An example of a spectrum file is shown below, with just the first few lines showing:

```
<negainspectrum(1,0)>   // signature
61                      // npoints
1.4269e+018             //  [1/cm3] Carrier density
//------------------------------------------------------------
//   lambda(nm)   gain(1/cm)
//------------------------------------------------------------
         780     -2389.84
         781     -2272.65
         782     -2162.16
         783      -2056.8
```

```
        784     -1955.47
        785     -1857.43
        786     -1762.15
        787     -1669.31
        788     -1578.72
        789     -1490.24
        790     -1403.77
        791     -1319.23
{continues}
```

The first line of the file must contain the signature:

```
<negainspectrum(1,0)>
```

The second line contains the parameter npoints – the number of points in the spectrum. The third line contains the carrier density at which this spectrum was generated, in units of [1/cm^3]. If you are using quantum well materials then this carrier density is the 2D carrier density divided by the total thickness of the quantum wells.

After this follows the spectrum data in two columns as shown in the example above.

Three comment lines (beginning //) must be included before the spectrum data as shown.

You should arrange your spectrum files so that they will naturally order in increasing carrier density. A convenient way of doing this is by adding a number representing the current density or carrier density to the end of the filename. A suitable example filename set might be:

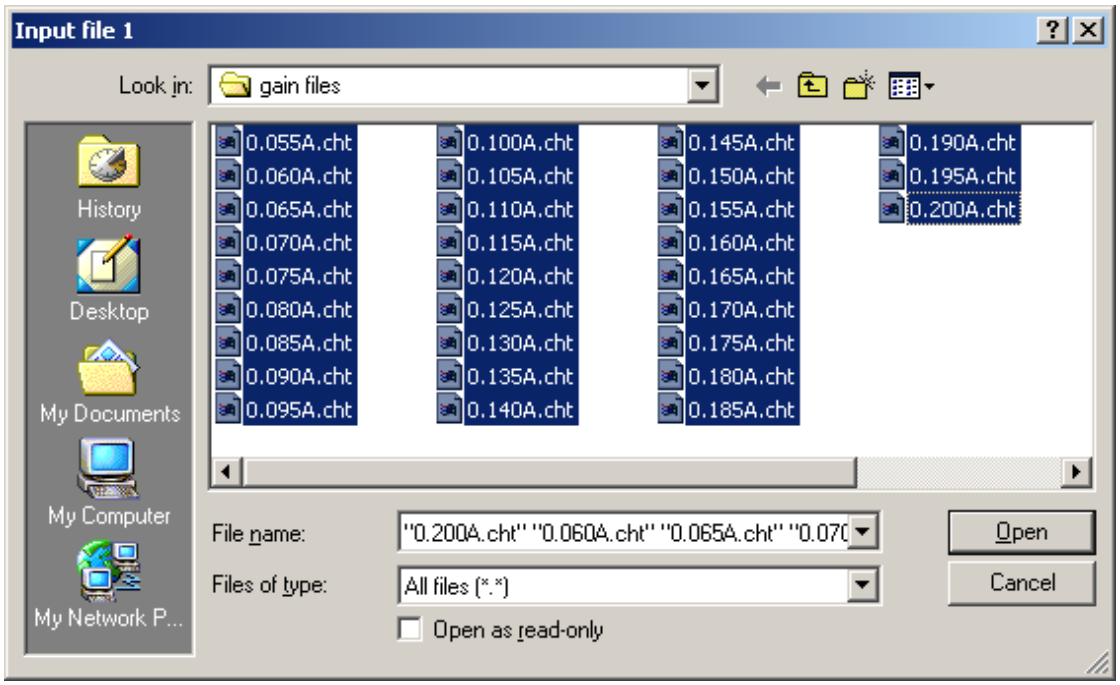
```
ingaasp-device3a8-spectrum-005
ingaasp-device3a8-spectrum-010
ingaasp-device3a8-spectrum-015
ingaasp-device3a8-spectrum-020
etc.
```

The spectrum files that you want to import into the Generator must all be in one directory.

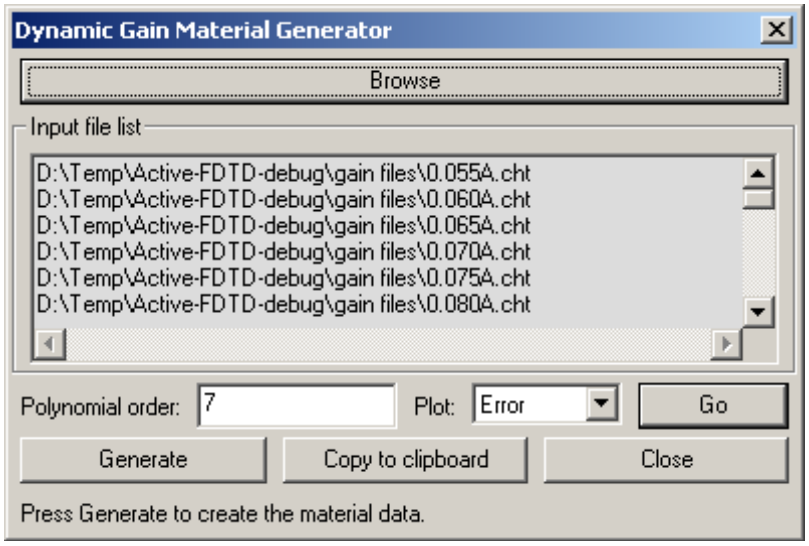
3.2 Operating the Generator

Here is the recommended sequence for producing a gain polynomial set for the Active-FDTD Module.

- Click on **Browse** and select all the spectra files in the file selector dialog box. To select multiple files use **Shift** together with the mouse or cursor keys. When you are ready you should see something like:

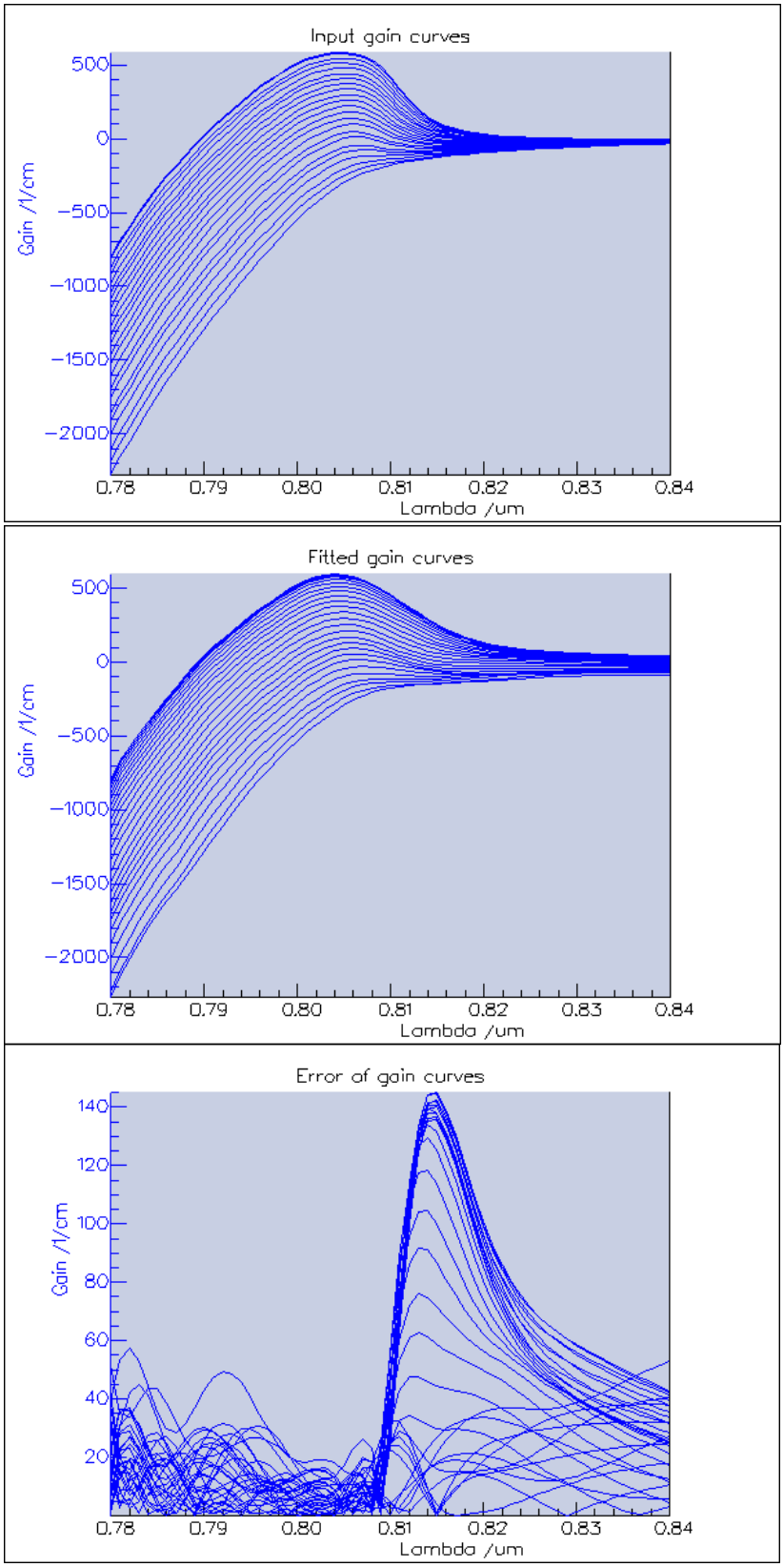


- When you are ready, click **Open**. You should now see all the files listed in the Generator panel:



- Next click on **Generate**. When it is done, a message will appear at the bottom of the Generator. You can now inspect the input and fitted spectra.
- Select one of Fitted Data / Input Data / Error from the Plot field and click **Go**. You will see a set of superimposed spectra representing either your input files, the fitted spectra or the difference (error) between the two.

If you are not happy with the fitted spectra you can try increasing the Polynomial order and regenerating another set of fitted spectra. Note that you will never get an exact match to the input spectra but the peak amplitude and position should be reasonably well retained. Below is a typical set of input, fitted and error spectra for a GaAs/GaAlAs QW material.



Above: showing a set of input gain spectra (top) computed externally by a quantum well gain model, a set of fitted spectra (middle) and the error between the two (bottom).

- Once you have a satisfactory set of fitted gain spectra you need to insert the data into a *material database* (.mat) file. Click on **Copy to Clipboard**. The polynomial data will be copied to the clipboard producing text something like:

```
// Define dynamic gain material
BEGIN GainMat(x) // material name
GAIN_4LORENTZPOLY 5 0.1 1.4967e+18 2.9225e+18 // polyOrder confFactor minCarrierDen maxCarrierDensity
-9.886699e+004  1.901265e-013 -1.618943e-031  6.839541e-050 -1.374988e-068  1.024834e-087
 1.145633e+004 -2.615417e-014  2.512867e-032 -1.269111e-050  3.411132e-069 -3.760714e-088
 3.021193e+004 -8.292523e-014  8.408895e-032 -4.092770e-050  9.659714e-069 -8.893744e-088
 1.519451e+007 -3.444592e-011  3.090054e-029 -1.371598e-047  3.013609e-066 -2.623142e-085
-3.914883e-012  9.166912e-030 -8.366254e-048  3.735285e-066 -8.127482e-085  6.893693e-104
-7.651717e-012  1.723831e-029 -1.521964e-047  6.634501e-066 -1.431751e-084  1.225654e-103
 2.504899e-012 -6.220944e-030  6.100064e-048 -2.921481e-066  6.864588e-085 -6.343753e-104
 3.105028e-011 -7.548992e-029  7.190121e-047 -3.354549e-065  7.679379e-084 -6.913431e-103
 2.843613e+015 -8.399532e-004  7.344051e-022 -3.208183e-040  6.748043e-059 -5.371914e-078
-6.905673e+014  6.763851e-003 -5.988481e-021  2.629413e-039 -5.725376e-058  4.946489e-077
 1.247437e+015  2.599427e-003 -2.336340e-021  1.038144e-039 -2.270152e-058  1.951533e-077
 6.319164e+022 -1.433080e+005  1.285994e-013 -5.709910e-032  1.254919e-050 -1.092638e-069
END
//----------------------------------------------------------------------
```

- This clipboard text may now be inserted into your material database file as described in §2.1 above.

3.3 Recommendations for an accurate fitting

You will find below a few recommendations to obtain an accurate fitting.

- Your range of carrier density should have quite a few data points in the region where the carrier density is below threshold. If you are using Harold to generate the gain curves, you can reduce the carrier density at which the regularly sampled data starts by modifying the parameters for *Voltage Bias* (accessed via the button **Options** in the Simulator). You can start by reducing **VoltageInc**, if need be you can also increase **switchV**.
- Do not hesitate to remove a few data curves of lowest carrier density as they may reduce accuracy of fitting (some trial and error may be needed).
- The fitting algorithm needs each data curve needs to start with absorption (negative value of the gain) at the smallest wavelength of the range. In Harold, you can get around this issue by adapting the wavelength range.
- For the choice of the polynomial order, some trial and error may be needed. We recommend that you start with a polynomial order of 5.

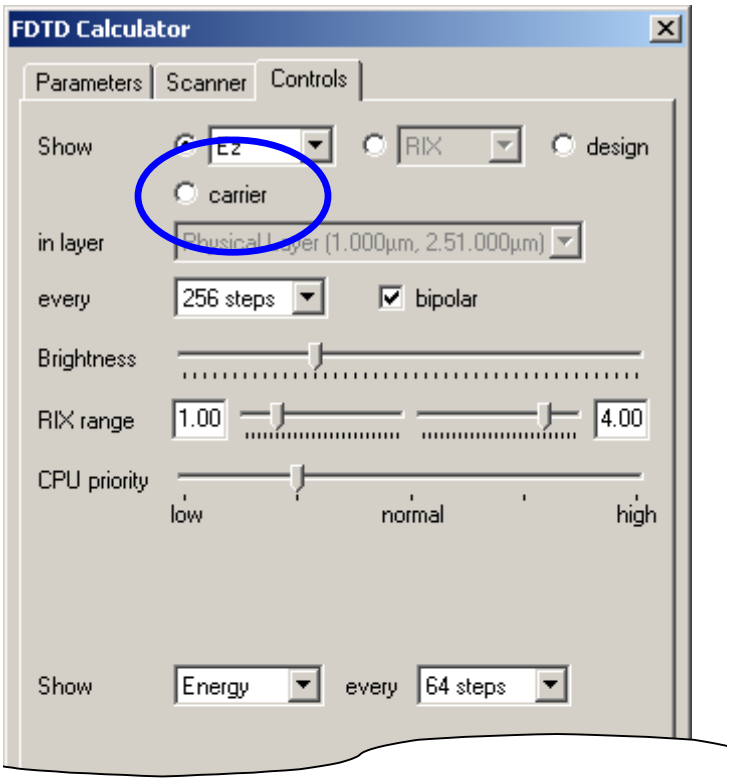
Chapter 4

Simulation Results and Monitor

The Active-FDTD Module provides a variety of additional interfaces for monitoring the carrier density profile.

4.1 Run-Time Monitoring

The FDTD Calculator's Control panel has one additional option as shown circled in blue in the figure below.

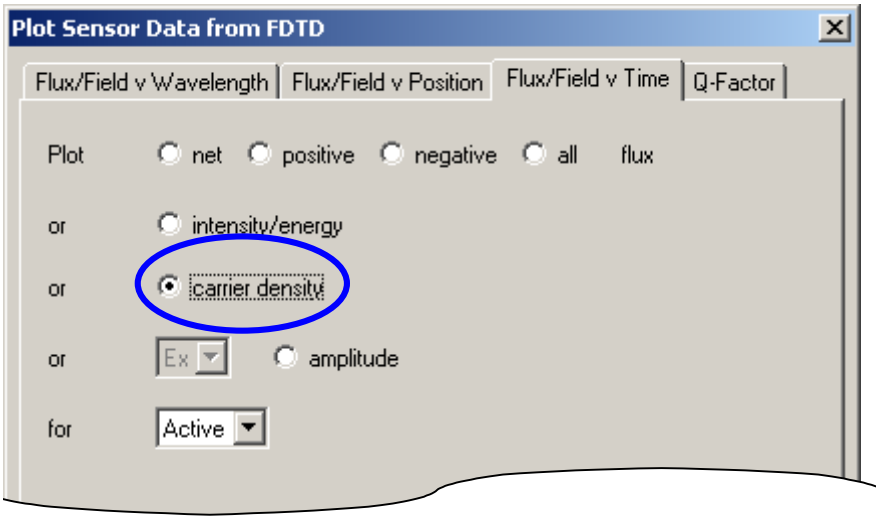


If you select this then the Device window will show the carrier density profile in the XZ plane. If it is 3D and you have multiple active layers then you must choose the active layer from the choice box provided. You can adjust the brightness with the control provided.

Remember that the carrier density resolution will be coarser than the optical grid resolution by the **CarrierSpatialResolution** factor.

4.2 Time-Resolved Carrier Density

Once a simulation is finished, you can plot the carrier density at the centre of a Sensor as a function of time. This is shown in the figure below.



4.3 Measuring Absolute Power

Calculating power based on the flux through a sensor in the time domain requires special attention from the user if he runs a 2D simulation. In a 2D simulation the measured power is actually a *power per unit height*. (Despite the fact that the plot says the unit is [W], in fact it is [W/m].)

In 2D the simulator does not know about the shape of the optical mode in the vertical (y) direction. However the total power W, integrated over the vertical direction is given approximately by:

$$W = P \cdot \textbf{TrueActiveThickness / confinementFac2D}$$

where P is the power per unit height as reported by the 2D sensor and the other parameters are as set in the FDTD Calculator properties panel. Thus you always need to apply this equation to the sensor power in order to obtain the absolute power exiting the laser.

Chapter 5 General Notes

There are a few things that you should be aware of or be careful about when using the Active-FDTD module. This section covers matters that do not fit elsewhere in this manual.

5.1 Spontaneous Emission

As stated previously, this algorithm does not model the spontaneous emission process. To start lasing you will need to inject light into the cavity from e.g. a dipole source, with a finite power at the resonant wavelength of the cavity. You can either use a pulse or for a more realistic source you could use a CW incoherent source. The CW incoherent source has a more or less constant intensity with time but achieves the necessary spread of optical frequencies by the random phase variations – see main application manual for more details of the incoherent dipole.

5.2 Carrier Dynamics

In a typical semiconductor material used in optical devices, the spontaneous lifetime of the carriers is around the 1 to 10ns range, decreasing with higher density. This is obviously far too large for a realistic FDTD simulation where the time step might be 0.05fs and which would typically run for a few 10's or 100's of ps, maybe 1ns at best. Thus it is necessary to speed up the carrier dynamics to better match the timescales of an FDTD simulation. We can do this using the **CarrierAccelFactor**. This works simply by increasing the right-hand side of the carrier rate equation (see introduction):

$$\frac{d\rho}{dt} = \left(-S - \frac{\rho}{\tau(\rho)} + \frac{J}{ed} \right) \times CarrierAccelFactor$$

Note that this will not affect the steady state, so that the carrier density distribution and optical power will all be the same as if the acceleration factor had not been used.

You can usually use quite large values of factor – 1000x or more. The guiding principle- is that you must not reduce the carrier lifetime to a value shorter than the photon lifetime in your optical cavity.

Instead of using the **CarrierAccelFactor** you can alternatively artificially reduce the carrier spontaneous lifetime and increase the current injection density by a corresponding factor. The resulting steady-state optical output power will however have increased by the factor you increased the current by.

5.3 Instabilities

If your **CarrierAccelFactor** is too large you may observe instabilities or unphysical dynamics – see above. If you observe instabilities but still need a large acceleration factor to keep your simulation short, you can often reduce or eliminate the instabilities by setting a gain non linearity coefficient – see GAIN_EPS above.

5.4 Negative Carrier Density

You can easily enter a negative current in a source. If this is of sufficiently short duration then it will be of no consequence and there may be legitimate reasons to do so.

In a real device if you applied a negative current for too long then the diode voltage would quickly go negative as the carriers are depleted, so reducing the current eventually to zero in a typical setup. This model does not have a proper diode voltage model so that a negative current can cause the carrier density to go negative. If this occurs the model will re-set the carrier density to zero but obviously your simulation will not be realistic.

Index