
This report outlines the most recent updates and enhanced capabilities of Translight. An example of the new capability of the code has been included which can be used as a tutorial in conjunction with “Tutorial 1: Omni-directional reflectance from a Bragg stack using Dmitry Chigrin’s structure.” on page 35.



3.1: Updates to Translight Code

In response to issues reported as a result of a distribution of a PC based version of the software and to the needs of others several updates have been made to the source code.

Band Structure Calculation

While Translight has been developed to compute the transmission and reflection coefficients for complex photonic crystals it can also perform elementary band structure analysis by calculating the Eigen values of the real space transfer matrix for a simple cell. As this calculation is limited to a single cell analysis. Consequently the output for complex cells and crystals that are built with more than one file using templates will reflect the band calculation for the first cell only. The visual interface for the windows version of the code does not allow activation of this feature, the band structure calculation is activated through the control file located in the same directory as the main Translight executable. Versions earlier than v2.00 of Translight did not support the single cell band structure calculation.

Coating Activation

Within Translight the user can choose to apply coatings to cylinders and spheres. Individual features can be coated with different materials by using crystal templates. At the current time the coating thickness is defined as a percentage of the radius of the cylinder or sphere. If the user activates the coatings option within the user interface then all of the features will be coated.

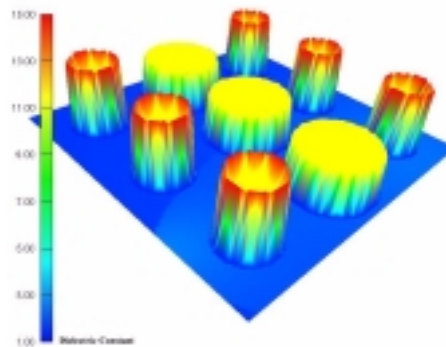


Figure 3.1 Example of Coating Routine Capabilities

Within the virtual reality output files coatings are represented as semi translucent skins over a solid shape of the same colour.

Mesh Building & On Screen Report Progress Visual Feedback

For all feature types, cylinders, spheres, rotated bars and bars, while the crystal is being built on screen progress indicators display the progress as each cell is built. If dispersive models for materials or loops over materials or feature sizes are activated then the progress will also be shown each time the crystal is rebuilt. Once the crystal is built the code embarks upon the analysis of the crystal. The user can choose whether or not on screen reporting is activated during the analysis phase, the on screen reporting is automatically activated for true three dimensional structures. This has been done as substantial amounts of time can be required for three

dimensional structures such as the woodpile or colloids, and the on screen reporting indicates that the program has not crashed.

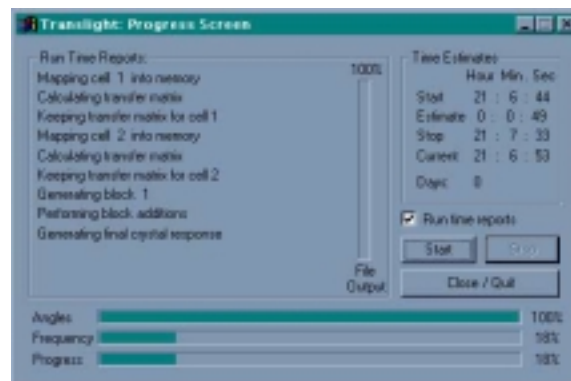


Figure 3.2 Example of On Screen Reporting During a Calculation

The user interface now also a few lattices demonstrated in pictorial form at the bottom of the templates tab. These pictures have no functionality and are aesthetic updates only.

Calculation Time Estimation

The progress screen now also includes an estimation as to the number of days that a calculation will take. For most 1D and 2D scenarios for a sensible number of frequency points, loops and incidence scans the program will complete execution in a range from a few seconds to several minutes or hours. For 3D systems the calculation time is considerably larger and extra care should be taken when setting up the calculation. For the pre-defined Opal and woodpile or layer by layer systems the calculation run times can run to several days.

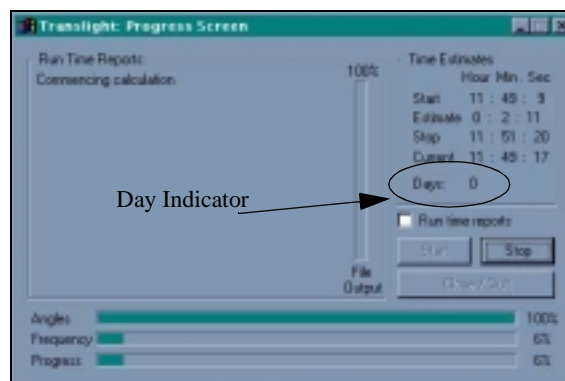


Figure 3.3 Updated Progress Screen with Day Estimation Included.

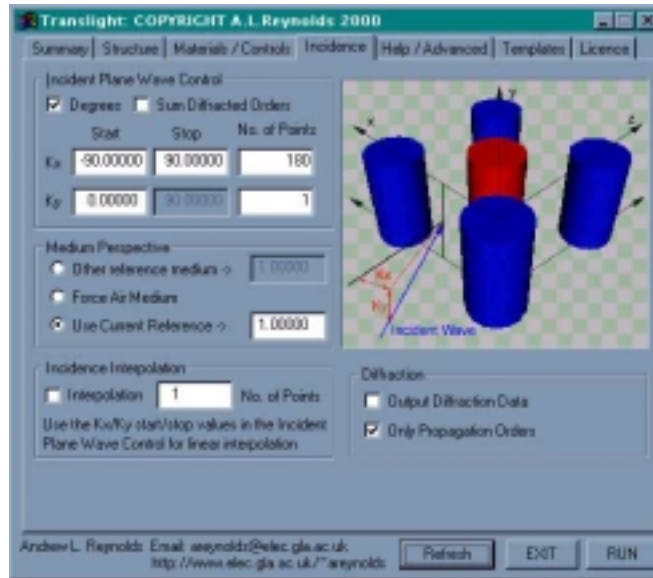


Figure 3.4 Incidence Control Tab with negative angle capability.

Negative Angular Incidence Control

The windows GUI now also accepts negative incidence angles within the Incidence control tab. Negative incidence angles could be set through the control file but now can be set on screen without any issues¹. Angles can be set in degrees or in k-vector depending on the user's settings.

Diffracted Orders

Controls have been added to allow the user to output information concerning the diffraction through the photonic crystal. A further option has been added to allow propagating or non propagating order information to be included in the diffraction files. The diffraction information is sorted using polarisation and in terms of reflection and transmission. The files are output into the folder named 'Difrac' located in the programs root directory.

Medium Perspective

The user can now choose any ambient medium for the crystal. The ambient medium is now independent of the materials that are used to build the crystal itself. There are three options available, use the same material as the reference material for the crystal, force an ambient air medium or choose another user definable value. Which ever value is used it must be real and not imaginary.

1. A.L. Reynolds would like to thank to Dr. Paul Tapster for reporting this bug.

Rotated Bars

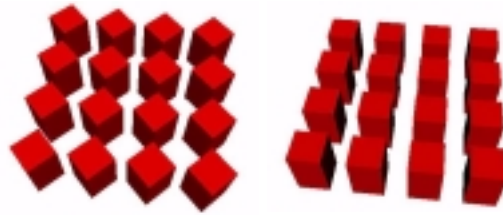


Figure 3.5 Rotated Bars Routines

Square cross section bars that have been rotated by *left: 45°*, *right: 0°*

The rotated bars function of the code has now also been activated. This allows the user to rotate all rectangular features within a lattice. Figure 3.5 shows square bars that have been rotated and not rotated. This aspect of the code is still unfinished as it is the intention that the rotation of each feature should be individually specified allowing the creation of more complex shapes such as crosses. Square cross section bars have been used for this example but the user is free to define a rectangular shape via the template files.

Bug Fixes & Elementary Updates

The original distribution of the code contained a pre-defined system which generates a close packed colloid system for analysis in the $\langle 111 \rangle$ direction. When the same library of the crystal is now used out with pre-defined system usage the code no longer enforces the close packed condition, the user is free to choose the radius of the spheres which make the crystal.

The code no longer crashes when the generate template button is pushed when the code is already using templates. Evidently the code cannot generate a skeletal template system while it is currently reading from the same files. The user is now notified of this situation and then returned to the user interface.



Figure 3.6 Warning feedback dialogue.

This screen is activated when the user tries to generate a template whilst the code is already using the template files, in versions earlier than v2.00 the code would crash in a similar circumstance.

3.2: Boundary condition & dissimilar discretisation mesh implementation.

This new feature of the code allows the use to use cells that have different dimensions and discretisation meshes in the z-direction, see Figure 3.7. To demonstrate this capability of the code the structure published by Smith et al. is examined². Figure 3.7 shows the structure that was studied, reproduced with kind permission from the original publication.

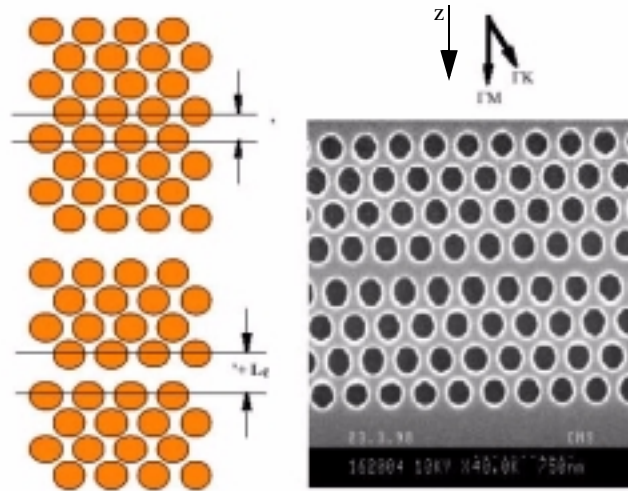


Figure 3.7 Smith's in plane microcavity structure.

The original diagrams as used by Smith defining the microcavity length L_c , (left), and a SEM picture of the structure with a 70nm cavity introduced in the centre of the crystal (right) in conjunction with the definitions of the main crystal directions ΓM , ΓK and the propagation direction $z=\Gamma M$.

The structure studied is a hexagonal arrangement of air holes in an AlGaAs substrate. The authors report that the substrate has an effective refractive index of $n_{\text{eff}}=3.32$ ($\epsilon_{\text{eff}}=11.0224$, a filling fraction of the lattice is 30% and a lattice constant of 220nm. These figures can be used to generate a radius to lattice spacing ratio of $r/a=0.28754$. However we note that the effective index of the medium is somewhat lower than the actual value AlGaAs, assuming a low concentration of aluminium a refractive index value of $n_{\text{GaAs}}=3.50574$ ($\epsilon_{\text{GaAs}}=12.29$) was used in the following situations.

To generate the lattice in Figure 3.7 the template capability of Translight³ is exploited and several cells are used, each of the cells used are shown in Figure 3.8. As some of the cells are similar the code can be set to reuse certain cells such that to build the structure only five cell definitions are required to build a structure that requires nine individual cell elements.

2. C.J.M. Smith, T.F. Krauss, R.M. De La Rue, D. Labilloy, H. Benisty, C. Weisbuch, U. Oesterle, R. Houdre, "In-plane microcavity resonators with two dimensional photonic bandgap mirrors", IEE Proceedings Optoelectronics Part J, Special Issue on Photonic Crystals and Photonic Microstructures, Vol. 145, (6), DEC 1998
3. Translight is available as shareware software subject to acceptance of the terms and conditions of the licence at <http://www.elec.gla.ac.uk/~areynolds> in the software section. Manuals and tutorials are also available on line.

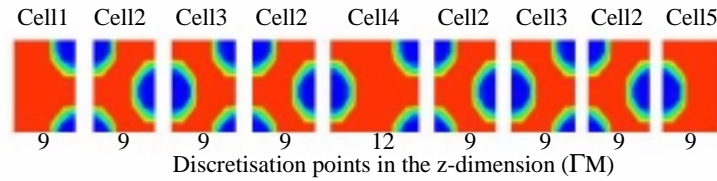


Figure 3.8 The cells used to build Smith's structure.

All of the cells with the exception of cell 4 have the same geometry and z-dimension. Figure 3.9 shows the template file used for cell 2 in Figure 3.8 above. The x-dimension of the cell reflects lattice spacing, 220nm, which has been discretized using 10 mesh points.

```
|----Description--|-----X-----|-----Y-----|-----Z-----|
Discretisation Mesh      10           1           9
Cell Dimensions (m) 0.2200E-06  0.2200E-06  0.1905E-06
Build Crystal Using: c
Lattice Constant:0.2200E-06
-----FEATURES-----
*No.*|***Material***|**OriginsMatrix**|AXIS|*Size*|*Coating**Value**|**VRML**|
*Type|-Real-|-Imag--|--X-----|--Y-----|--Z---|--|-R/A--|T/F-Real-|-Imag--|-Colour-|
1 r   1.000   0.000   0.000   0.000   0.000 y  0.2875 F  1.000  1.000  red
2 r   1.000   0.000   1.000   0.000   0.000 y  0.2875 F  1.000  1.000  red
3 r   1.000   0.000   0.500   0.000   1.000 y  0.2875 F  1.000  1.000  red
```

Figure 3.9 Template files used to build cell 2.

To introduce the planar microcavity into the system the z-dimension of the cell 4 should be increased by 70nm. The template file for cell 4 is shown in Figure 3.10.

```
|----Description--|-----X-----|-----Y-----|-----Z-----|
Discretisation Mesh      10           1          12
Cell Dimensions (m) 0.2200E-06  0.2200E-06  0.2605E-06
Build Crystal Using: c
Lattice Constant:0.2200E-06
-----FEATURES-----
*No.*|***Material***|**Origins Matrix**|AXIS|*Size*|*Coating**Value**|**VRML**|
*Type|-Real-|-Imag--|--X-----|--Y-----|--Z---|--|-R/A--|T/F-Real-|-Imag--|-Colour-|
1 r   1.000   0.000   0.500   0.000   0.000 y  0.2875 F  1.000  1.000  red
2 r   1.000   0.000   0.000   0.000   1.000 y  0.2875 F  1.000  1.000  red
3 r   1.000   0.000   1.000   0.000   1.000 y  0.2875 F  1.000  1.000  red
```

Figure 3.10 Template file for the microcavity cell 4.

Both the z-cell dimension size and the number of points used to discretize the cell have been increased to introduce the planar microcavity into the structure. This file demonstrates the power of the template files and the normalised co-ordinate system used to place each element. The user is free to alter the z-dimension size and the cell size and the features will always be placed at the same relative point in the cell so that cell size alterations do not necessitate position changes.

Once defined the code can be used to generate a chunk of the crystal which is shown in Figure 3.11 next to the original SEM of the structure.

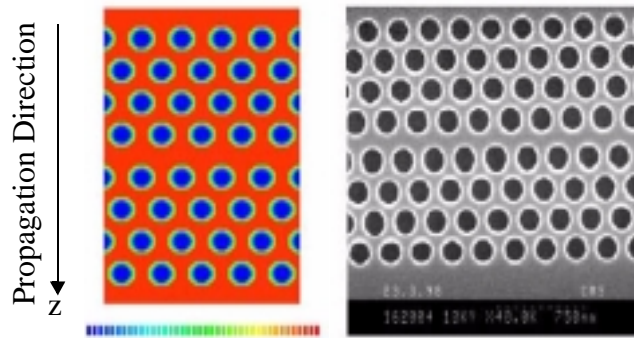


Figure 3.11 Computer Generate structure vs SEM

Left: The computer generate structure made by Translight. Right: The original SEM of Smith's structure. Both structures have a 70nm microcavity introduced into the crystal which can be accurately modelled by Translight by exploiting the dissimilar mesh and z-cell size capability of the code.

Modelled vs Actual TE Transmission Response.

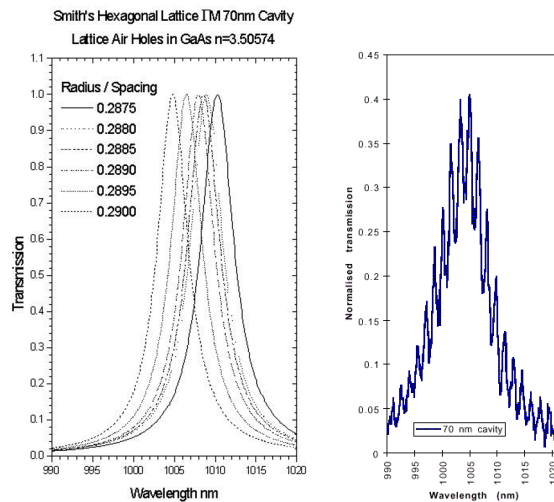


Figure 3.12 Calculated and measured response of the 70nm microcavity.

As the filling fraction of the actual lattice has been estimated several hole sizes were tried until the predicted response was matched with the measured response. Inaccuracy in the actual cavity size will also tune the defect response, but using the assumed cavity size of 70nm a radius to lattice spacing ratio of $r/a=0.2895$ gave the correct response. This corresponds to a filling fraction of <calculate the filling fraction>





3.3: UNIX Code Issues

Capabilities

A UNIX version of Translight has been installed on both HP, Sun and IBM SP2 Unix systems. This version of code is fully specified and has the same capabilities of the windows version. For the UNIX installation the source code is available and can be modified. It is recommended that before any modifications are made to the source code that the a complete copy of the program source files is made. The UNIX version of the code can be used in conjunction with a powerful drag and drop interface systems which allows version control and alternative make files to be used. This is especially useful as one make file can be used to debug the code and another to build the release version.

Version Control / Directory Structure

Several directories are required for the drag and drop system to function. From the root directory the following directories are required for correct successful operation of the code.:


















Root Directory:	Contains / Description	Contains / Description
<p>Contains:-></p>  src	<p>For example:</p>  a01  b03  w01	<p>Each of these folders contains the source code related to that particular version or program type.</p>
<p>Contains:-></p>  bin	<p>Contains the binaries for each version of the code. The file 'pgm.go' is also held here which is the shell script that runs the code from the drag and drop interface. The shell script determines which version of the code is to be built by choosing the correct make file is used and which prefix to the make file is employed.</p>	
<p>Contains:-></p>  mak	<p>Contains several make files: opalB.mak, opalW.mak and opalD.mak. Each make file identifier (B, W, D) is prefixed with 'opal'. This prefix can be set through the 'pgm.go' script located in the 'bin' directory.</p>	
<p>Contains:-></p>  newres	<p>For example:</p>  a01Dtest  b03Bhex  w01Wopal	<p>Each folder contains:- The first three characters identify which version of the code within the 'src' directory that will be used. The next letter identifies which make file that will be used from the 'mak' directory. Thereafter the use is free to choose the label.</p> <p>  Coeff Coeff: Transmission and reflection coefficients sorted by TE and TM polarisation  Crystal Crystal: data files generated about the structure of the crystal including the virtual reality file 'vrml.wrl'  Difrac Difrac: diffraction information for a crystal given that only a single frequency is analysed.  Phase .Phase: diffraction phase information for a crystal given that only a single frequency is analysed.  Fields Fields: Under development.  Reports Reports: run time error files, debug files and run time information. Check here for errors.  Template Template: contains the cell files that are used to construct custom crystals. Refer to the manual for further information </p> <p>The control file 'control.dat' is also present at this level in conjunction with an executable 'wipedata'. Wipedata erases all of the data files within each of the sub directory folders other than the data files within the Template subdirectory. The time estimate file is also stored at this level.</p>

TABLE 1 :Directory structure for a UNIX installation of Translight.





Contains:->  obj	This folder contains all the complied 'objects' related to all of the versions of code. This directory should be periodically cleaned out when versions of code are being developed as old objects may cause problems with new builds of a version.
Contains:->  lst	Working directory for the code, the modules related to the code are stored in this directory. As with the objects directory this directory should be cleaned when new versions of the code are being developed to save compiler confusion.

TABLE 1 :Directory structure for a UNIX installation of Translight.

NOTE: If this directory structure is not present then the code will not function correctly and no error messages will be displayed informing you of the problem. Unlike the windows version the UNIX version does not self install. If serious problems persist with the installation please contact the developer.

Drag & Drop Interfacing

To use the drag and drop interfacing the user should set up an action on the HP desktop. If this is unfamiliar for the user contact your system administrator. The action should execute the following command:

```
/bin/csh/ -v /home/vis/bin/pgm.go %(File)Args%
```

The important aspect of this command is the '/home/vis/bin/pgm.go'. This should be tailored to reflect the correct path of the program and user installation. The action is used by dragging a folder within the 'newres' folder, such as 'w01Wopal' onto the created action icon. The action then passes the folder name and path to via the %(File)Args% to the script 'pgm.go' located within the 'bin' subdirectory. The script then processes this information and calls the relevant make file and batches the binary for execution.

Using Templates in the UNIX version

Templates are used by changing the 'SYSTEMNO' in the Control.dat file to be a negative number. Thereafter template usage is the same as the windows version, see the manual and "Tutorial 1: Omni-directional reflectance from a Bragg stack using Dmitry Chigrin's structure." on page 35 for further details.

Pre-defined Systems

There are thirty three pre-defined systems available within the UNIX version of the code. These are listed in Table 2⁴

System Number	Description of the system.
1	Ozby: 500GHz Woodpile, 8 Cells
2	Ozby: 94GHz Woodpile, 8 Cells
3	Generic Hexagonal Lattice, Gamma M
4	Generic Hexagonal Lattice, Gamma K
5	Labilloy's Hexagonal Lattice, Gamma M
6	Labilloy's Hexagonal Lattice, Gamma K
7	Close Packed Opal:<111>
8	Close Packed Opal:<100>
9	Generic Square Lattice: Gamma-X L=8a
10	Villeneuve's Square Lattice, 5 periods
11	Villeneuve's Square Lattice, planar defect
12	1 feature many cells, ZMesh & Aspect Control
13	Ward's Bragg Stack, 1D
14	Sigalas's Metallic System, 2D
15	Winn's Bragg Stack in Air
16	Winn's Bragg Stack in Dielectric
17	Chigrin's Bragg Stack in Air
18	Chigrin's Bragg Stack in Dielectric
19	Generic Square Lattice: Gamma-M L=8a
20	Square Drilled Bragg, Top Incidence
21	Square Drilled Bragg, Side Incidence
22	Hexagonal Drilled Bragg, Top Incidence
23	Hexagonal Drilled Bragg, Side Incidence
24	Hexagonal Gamma M, aspect controlled
25	Hexagonal Gamma K, aspect controlled
26	Hexagonal Gamma M, Defects 1 in 2
27	Hexagonal Gamma M, Defects 1 in 3
28	Hexagonal Gamma M, Defects 1 in 4
29	Hexagonal Gamma M, Defects 1 in 5
30	Hexagonal Gamma K, Defects 1 in 2
31	Hexagonal Gamma K, Defects 1 in 3
32	Hexagonal Gamma K, Defects 1 in 4
33	Hexagonal Gamma K, Defects 1 in 5

TABLE 2 :Pre-defined systems within the UNIX version of Translight.

4. There may be some differences in the predefined systems between the Windows and UNIX versions of the code.

