ECE 595, Section 10
Numerical Simulations
Lecture 28: S-Matrix Computations

Prof. Peter Bermel

March 22, 2013

S-Matrices: Periodic Solution Strategy

- Can use the H-fields in F-space

$$h(z) = e^{iqz}\left\{ \phi_x \hat{\mathbf{x}} + \phi_y \hat{\mathbf{y}} - \frac{1}{q}(\hat{k}_x \phi_x + \hat{k}_y \phi_y)\hat{\mathbf{z}} \right\}$$

- Eigenvalue equation becomes

$$\left\{ \begin{pmatrix} \hat{\eta} & 0 \\ 0 & \hat{\eta} \end{pmatrix} \left[q^2 + \begin{pmatrix} \hat{k}_x \hat{k}_x & \hat{k}_x \hat{k}_y \\ \hat{k}_y \hat{k}_x & \hat{k}_y \hat{k}_y \end{pmatrix} \right] + \begin{pmatrix} \hat{k}_y \hat{\eta} \hat{k}_y & -\hat{k}_y \hat{\eta} \hat{k}_x \\ -\hat{k}_x \hat{\eta} \hat{k}_y & \hat{k}_x \hat{\eta} \hat{k}_x \end{pmatrix} \right\} \times \begin{pmatrix} \phi_x \\ \phi_y \end{pmatrix} = \omega^2 \begin{pmatrix} \phi_x \\ \phi_y \end{pmatrix}$$

- More compactly represented as:

$$[\mathcal{H}(q^2 + K) + \mathcal{K}]\phi = \omega^2 \phi$$

- Where the eigenvectors ϕ have a unique orthonormality condition: $\phi_n^T(\omega^2 - \mathcal{K})\phi_{n'} = \delta_{nn'}$

S-Matrices: Periodic Solution Strategy

- Can rephrase:

$$[(\omega^2-\mathcal{K})\mathcal{E}(\omega^2-\mathcal{K})-\omega^2 K]\phi = q^2(\omega^2-\mathcal{K})\phi$$

- Where *H*-fields are written as:

$$\begin{pmatrix} h_x(z) \\ h_y(z) \end{pmatrix} = \sum_n \begin{pmatrix} \phi_{x_n} \\ \phi_{y_n} \end{pmatrix} (e^{iq_n z} a_n + e^{iq_n(d-z)} b_n)$$

$$h_{\parallel}(z) = \Phi[\hat{\mathrm{f}}(z) a + \hat{\mathrm{f}}(d-z) b]$$

- And where *E*-fields are given by:

$$\begin{pmatrix} -e_y(z) \\ e_x(z) \end{pmatrix} = \sum_n \begin{pmatrix} \hat{\eta} & 0 \\ 0 & \hat{\eta} \end{pmatrix} \left[q_n^2 + \begin{pmatrix} \hat{k}_x\hat{k}_x & \hat{k}_x\hat{k}_y \\ \hat{k}_y\hat{k}_x & \hat{k}_y\hat{k}_y \end{pmatrix} \right] \times \begin{pmatrix} \phi_{x_n} \\ \phi_{y_n} \end{pmatrix} \frac{1}{q_n} (e^{iq_n z} a_n - e^{iq_n(d-z)} b_n)$$

$$e_{\parallel}(z) = (\omega^2-\mathcal{K})\Phi\hat{q}^{-1}[\hat{\mathrm{f}}(z) a - \hat{\mathrm{f}}(d-z) b]$$

S-Matrices: Periodic Solution Strategy

- Interface matrix in WC notation:

$$\begin{pmatrix} \hat{f}_l a_l \\ b_l \end{pmatrix} = I(l,l+1)\begin{pmatrix} a_{l+1} \\ \hat{f}_{l+1} b_{l+1} \end{pmatrix} = \begin{pmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{pmatrix}\begin{pmatrix} a_{l+1} \\ \hat{f}_{l+1} b_{l+1} \end{pmatrix}$$

- Where: $I(l,l+1) = M_l^{-1} M_{l+1}$

$$= \tfrac{1}{2}\hat{q}_l \Phi_l^T(\omega^2 - \mathcal{K}_{l+1})\Phi_{l+1}\hat{q}_{l+1}^{-1}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

$$+ \tfrac{1}{2}\Phi_l^T(\omega^2 - \mathcal{K}_l)\Phi_{l+1}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

S-Matrix Construction: Recap

- We can relate the *s*-interface matrix to the *t*-interface matrix from before:

$$s^{(p)} = \begin{bmatrix} t_{11}^{(p)} - t_{12}^{(p)} t_{22}^{(p)\,-1} t_{21}^{(p)} & t_{12}^{(p)} t_{22}^{(p)\,-1} \\ -t_{22}^{(p)\,-1} t_{21}^{(p)} & t_{22}^{(p)\,-1} \end{bmatrix}$$

- Then iteratively construct next S-matrix via:

$$\begin{bmatrix} T_{uu}^{(p)} & R_{ud}^{(p)} \\ R_{du}^{(p)} & T_{dd}^{(p)} \end{bmatrix} = \begin{bmatrix} \tilde{t}_{uu}^{(p)} \left[1 - R_{ud}^{(p-1)} \tilde{r}_{du}^{(p)}\right]^{-1} T_{uu}^{(p-1)} \\ R_{ud}^{(p-1)} + T_{dd}^{(p-1)} \tilde{r}_{du}^{(p)} \left[1 - R_{ud}^{(p-1)} \tilde{r}_{du}^{(p)}\right]^{-1} T_{uu}^{(p-1)} \\ \tilde{r}_{ud}^{(p)} + \tilde{t}_{uu}^{(p)} R_{ud}^{(p-1)} \left[1 - \tilde{r}_{du}^{(p)} R_{ud}^{(p-1)}\right]^{-1} \tilde{t}_{dd}^{(p)} \\ T_{dd}^{(p-1)} \left[1 - \tilde{r}_{du}^{(p)} R_{ud}^{(p-1)}\right]^{-1} \tilde{t}_{dd}^{(p)} \end{bmatrix}$$

S-Matrices: Periodic Solution Strategy

- In WC's notation:

$$S_{11}(l',l+1) = (I_{11} - \hat{f}_l S_{12}(l',l) I_{21})^{-1} \hat{f}_l S_{11}(l',l)$$

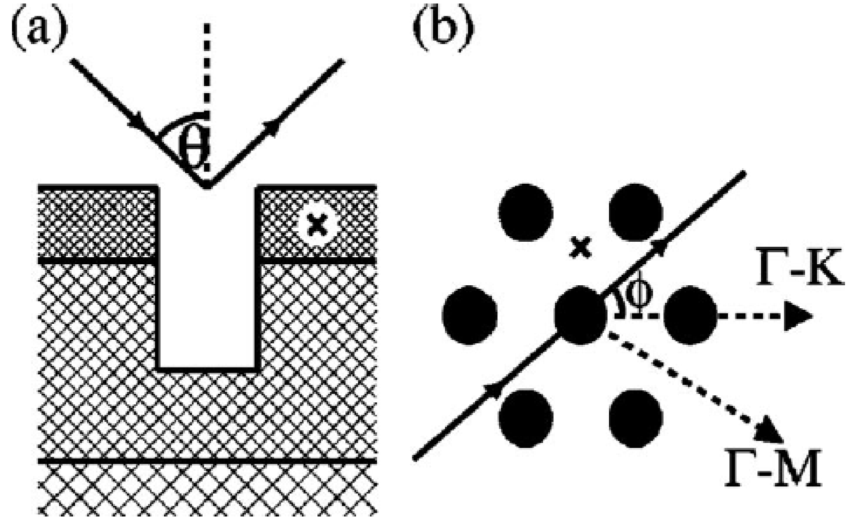
$$S_{12}(l',l+1) = (I_{11} - \hat{f}_l S_{12}(l',l) I_{21})^{-1}$$
$$\times (\hat{f}_l S_{12}(l',l) I_{22} - I_{12}) \hat{f}_{l+1}$$

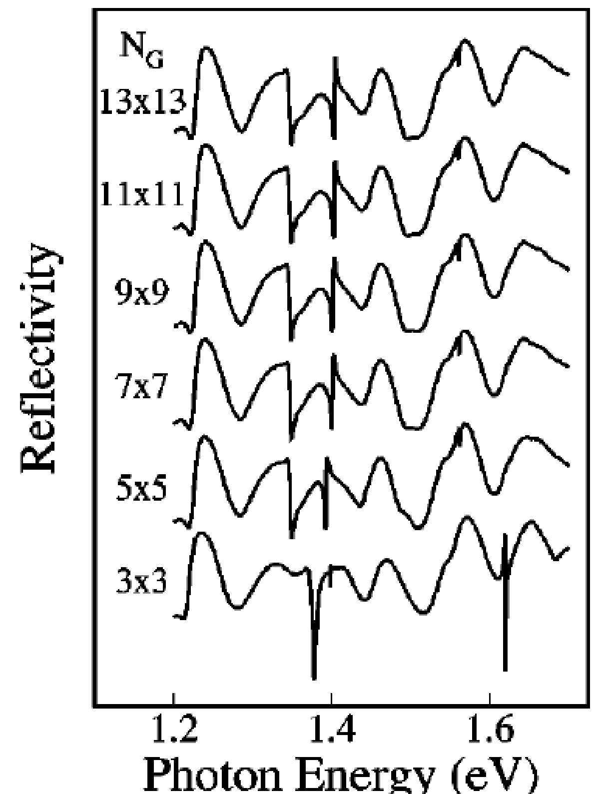
$$S_{21}(l',l+1) = S_{22}(l',l) I_{21} S_{11}(l',l+1) + S_{21}(l',l)$$

$$S_{22}(l',l+1) = S_{22}(l',l) I_{21} S_{12}(l',l+1) + S_{22}(l',l) I_{22} \hat{f}_{l+1}$$

S-Matrix Simulations

- Transmission through triangular lattice converges as number of plane waves N_G increases

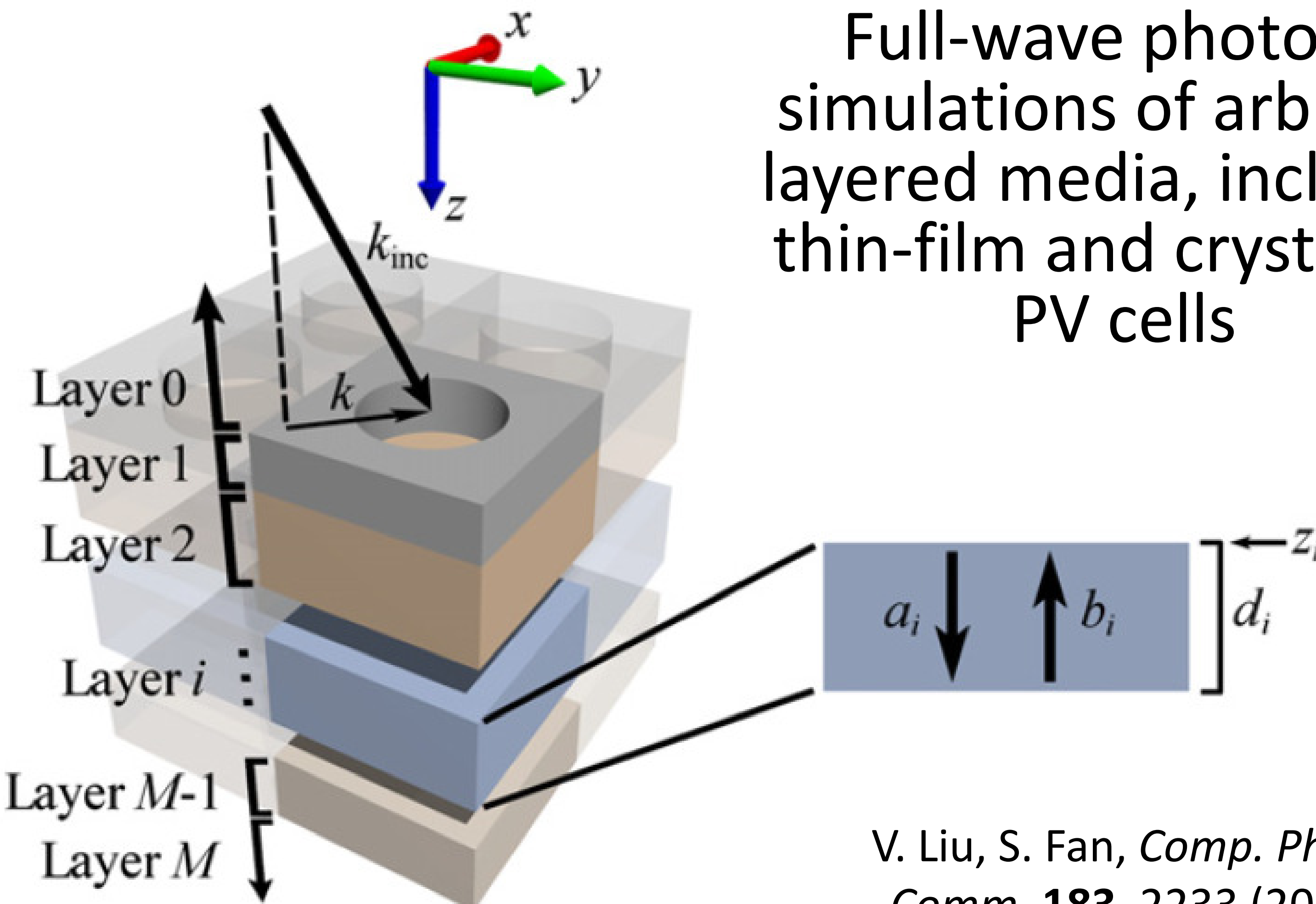




$$\varepsilon(\mathbf{G}) = \begin{cases} 2(\varepsilon_a - \varepsilon_b)\beta J_1(G\rho)/(G\rho) & \mathbf{G} \neq 0 \\ \varepsilon_a \beta + \varepsilon_b (1-\beta) & \mathbf{G} = 0, \end{cases}$$

Whittaker & Culshaw, *Phys. Rev. B* **60**, 2610 (1999)

Photonic Simulations with S^4

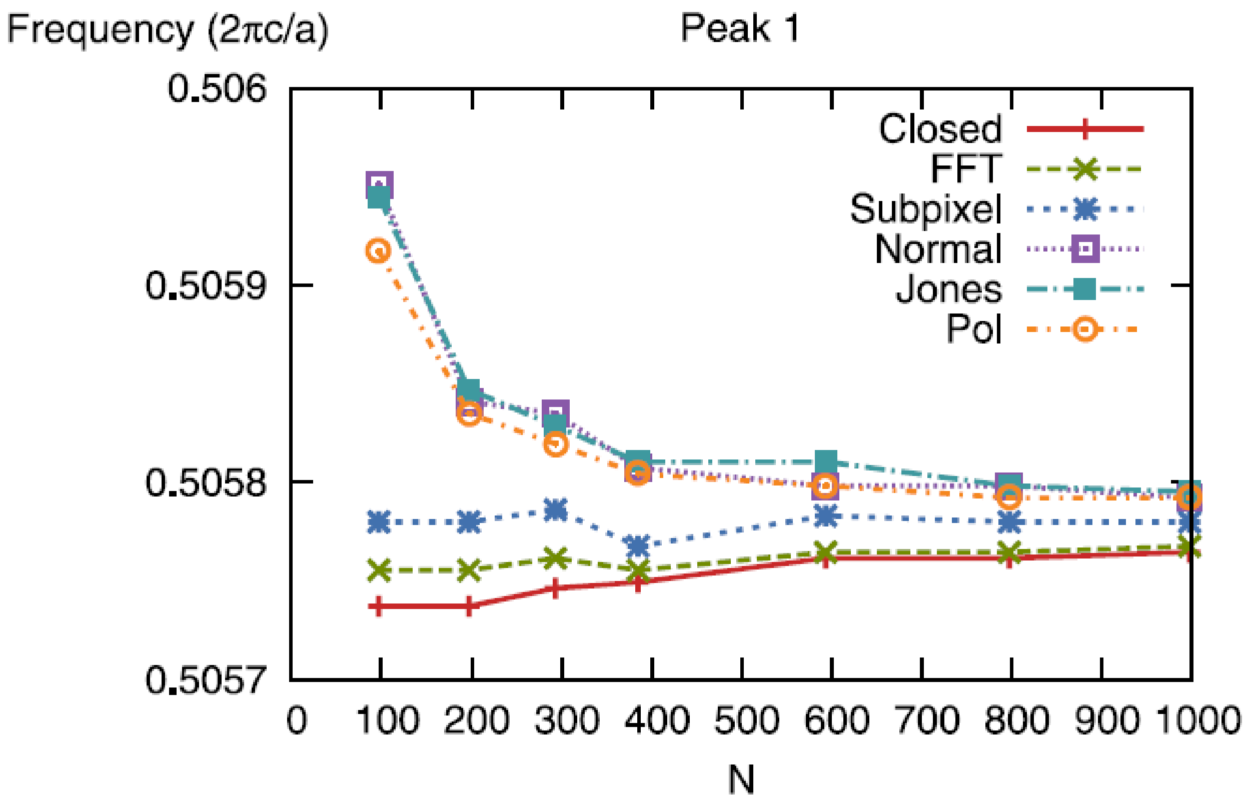


Full-wave photonic simulations of arbitrary layered media, including thin-film and crystalline PV cells

V. Liu, S. Fan, *Comp. Phys. Comm.* **183**, 2233 (2012)

Photonic Simulations with S^4

Accuracy improves systematically with computing power



V. Liu, S. Fan, *Comp. Phys. Comm.* **183**, 2233 (2012)

S^4: Lua Control Files

- Obtain a new, blank simulation object with no solutions:

```
S = S4.NewSimulation()
```

- Define all materials:

```
S:AddMaterial('name', {eps_real, eps_imag})
```

- Add all layers:

```
S:AddLayer('name', thickness, 'material_name')
```

- Add patterning to layers:

```
S:SetLayerPatternCircle('layer_name', 'inside_material',
{center_x, center_y}, radius)
```

S^4: FMM Formulations

- Specify the excitation mechanism:

```
S:SetExcitationPlanewave(
        {angle_phi, angle_theta}, -- phi in [0,180), theta in [0,360)
        {s_pol_amp, s_pol_phase}, -- phase in degrees
        {p_pol_amp, p_pol_phase})
```

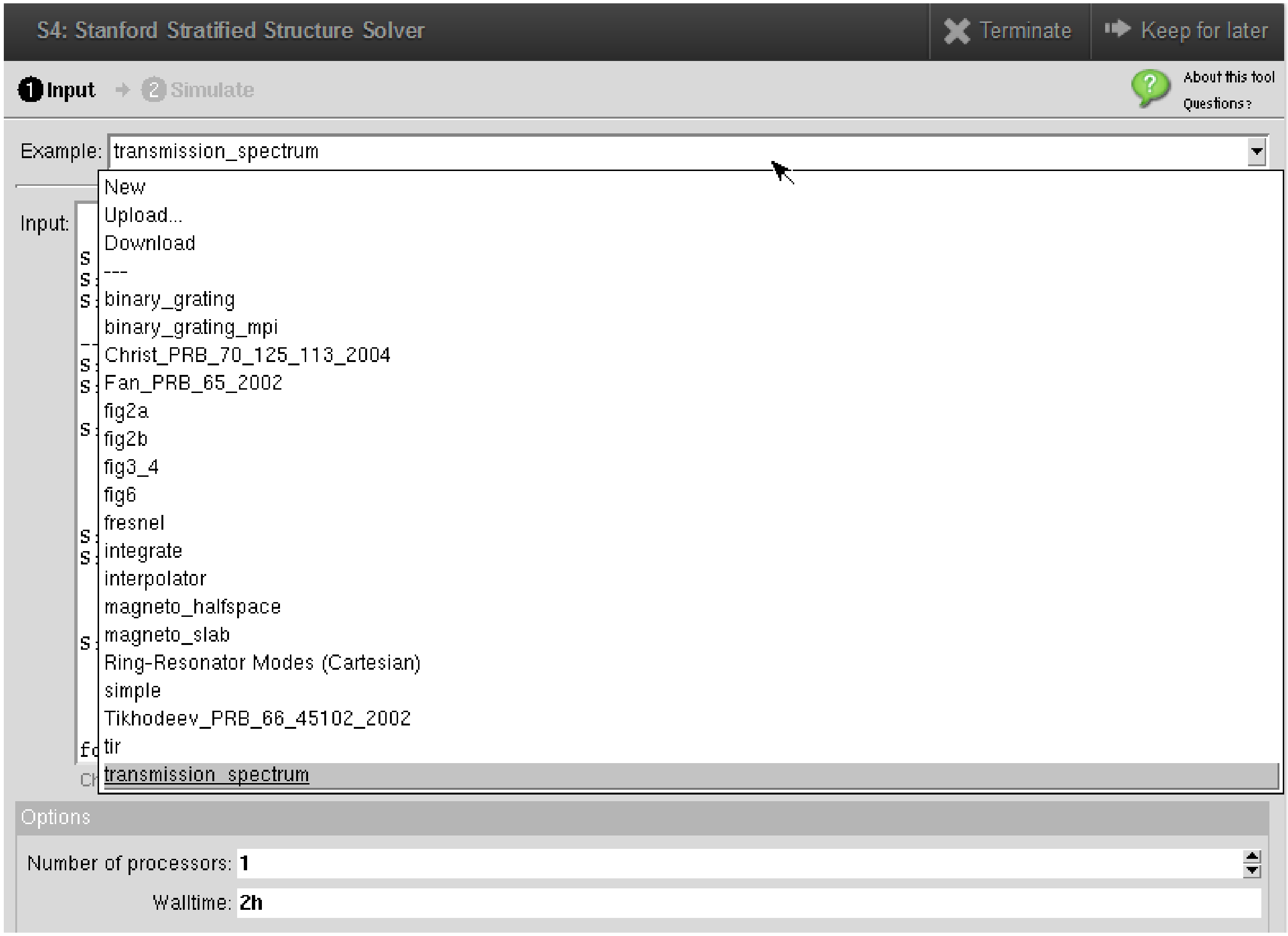
- Specify the operating frequency:

```
S:SetFrequency(0.4)
```

- Obtain desired output:

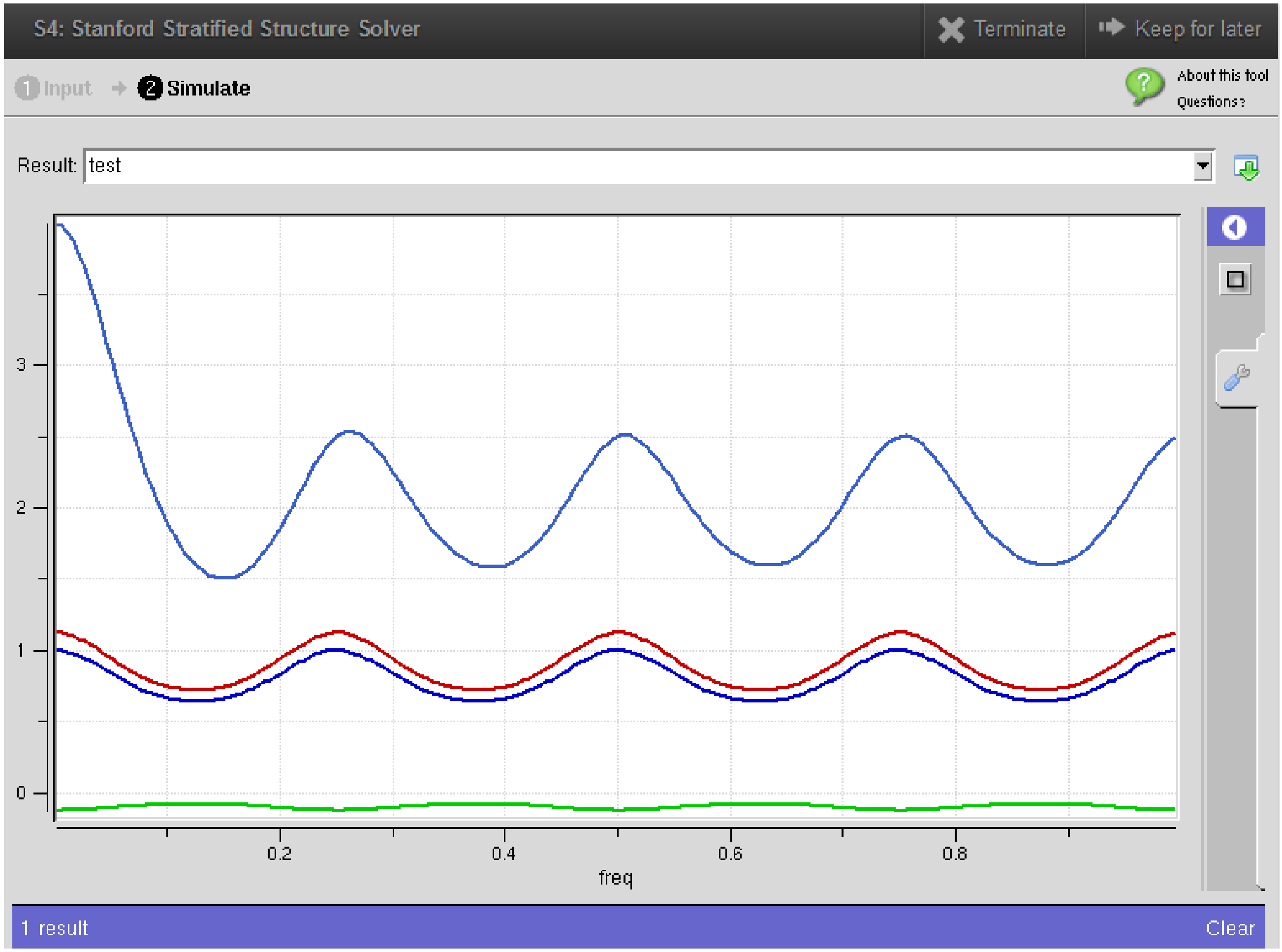
```
forward_power, backward_power = S:GetPoyntingFlux('layer_name', z_offset)
print(forward_power, backward_power)
```

S^4: Input



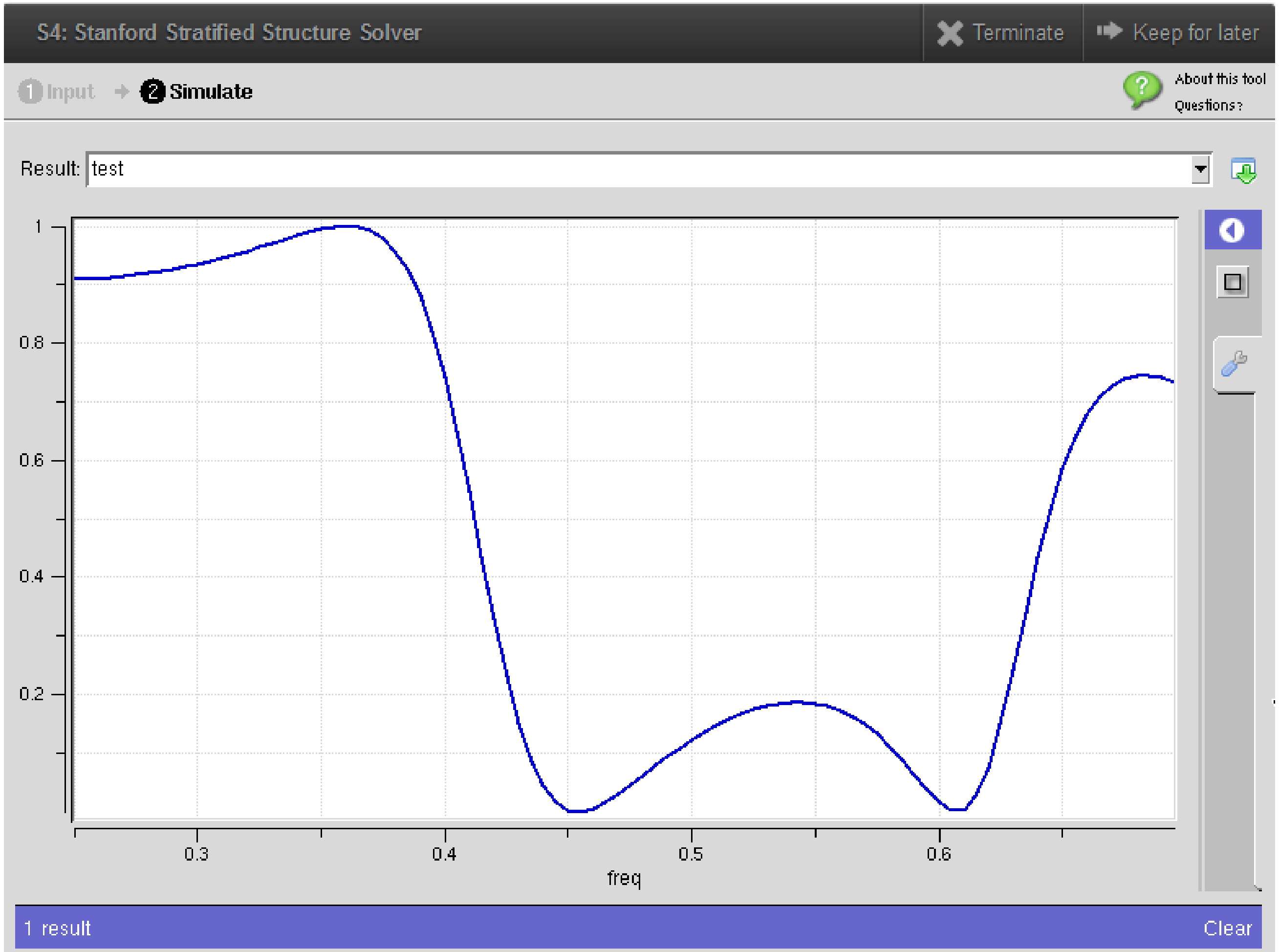
Can choose several examples drawn from the literature

S^4: Output



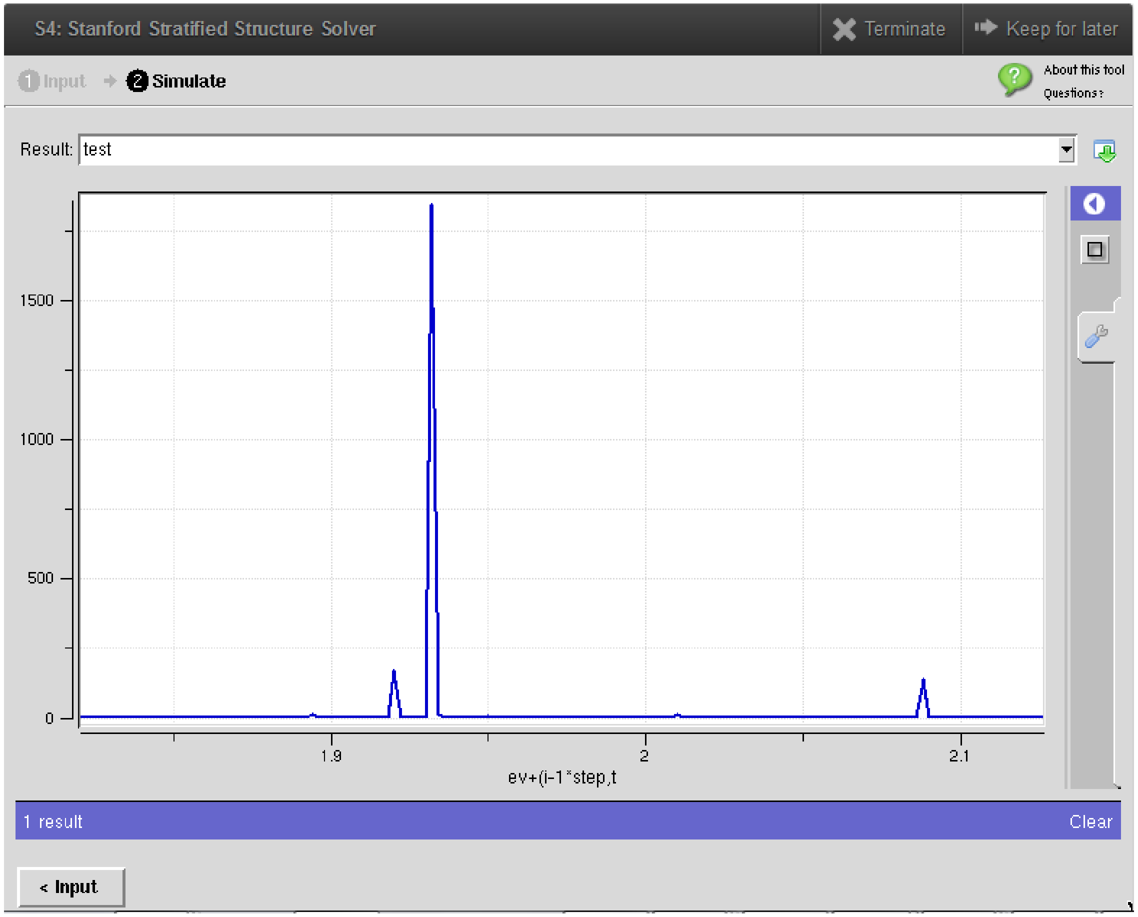
Transmission through multilayer stack matches analytical expression

S^4: Output



Transmission through 1D square grating of silicon and air

S^4: Output



- Transmission from Fig. 4 of Tikhodeev *et al.*, *Phys. Rev. B* **66**, 045102 (2002).

Next Class

- Is on Monday, March 25
- Will discuss CAMFR, see: http://camfr.sourceforge.net