

Efficient and Accurate BPM Simulation of a Polarization Converter-Splitter Based on Silicon Photonics

Introduction

A polarization converter-splitter is inherently a long waveguide device, based on adiabatic mode coupling [1]. Designing and simulating such large structures using rigorous FDTD (Finite-Difference Time-Domain) requires significant computer memory and extremely long computation times.

Because the structure varies slowly and exhibits minimal backward reflection, this type of long device is well suited for simulation using BPM (Beam Propagation Method). Since polarization coupling is involved, a full-vector BPM must be used [2].

BPM relies on several assumptions and may encounter stability issues, particularly for full-vector BPM applied to high-index-contrast structures such as those used in silicon photonics. This technical challenge can be mitigated by re-normalizing the power using two simulation approach [3].

In this paper, we demonstrate how a silicon-based polarization converter-splitter can be efficiently and accurately simulated using BPM. Benchmarking with FDTD shows that BPM can achieve comparable accuracy while reducing computation time and memory usage by several orders of magnitude.

Structure

The example is taken from a published paper [1] and its schematic diagram, reproduced from the reference, is shown in Figure 1.

Geometric Parameters of a 220 nm Si-Wire Covered with SiN

$W_0 = W_{out} = 0.54 \mu\text{m}$	$L_{tp2} = 44 \mu\text{m}$
$W_1 = 0.69 \mu\text{m}$	$L_{tp3} = 1.87 \mu\text{m}$
$W_2 = 0.83 \mu\text{m}$	$L_{dc} = 7 \mu\text{m}$
$W_3 = 0.9 \mu\text{m}$	$L_{tpout} = 14 \mu\text{m}$
$W_4 = 0.405 \mu\text{m}$	Edge-edge separation of two output waveguides is $1.15 \mu\text{m}$
$W_{gap} = 0.15 \mu\text{m}$	Working wavelength is $1.55 \mu\text{m}$
$L_{tp1} = 4 \mu\text{m}$	

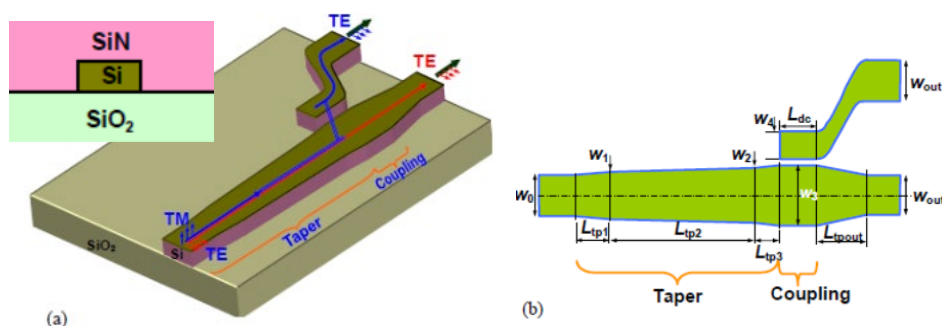


Figure 1. The schematic configuration of the proposed polarization splitter-rotator based on an asymmetrical DC. (a) the 3D view; (b) the top view.

Mode Calculation

Although the FemSIM FEM solver is more rigorous, the built-in BP mode solver [4] in RSoft CAD is sufficient for this simple waveguide structure. The calculated TE and TM modes are shown in Figure 2 and Figure 3, respectively. These modes are used as the input excitation at the input port and as overlapping targets at the output ports.

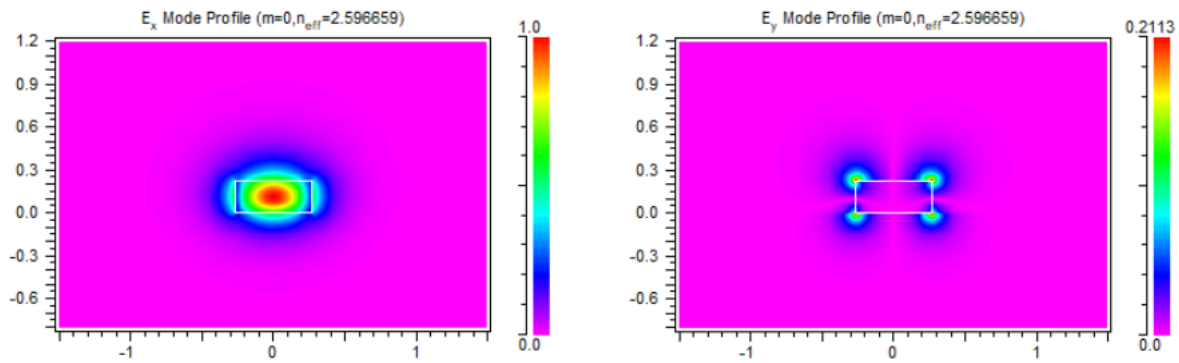


Figure 2. Calculated TE full-vector mode

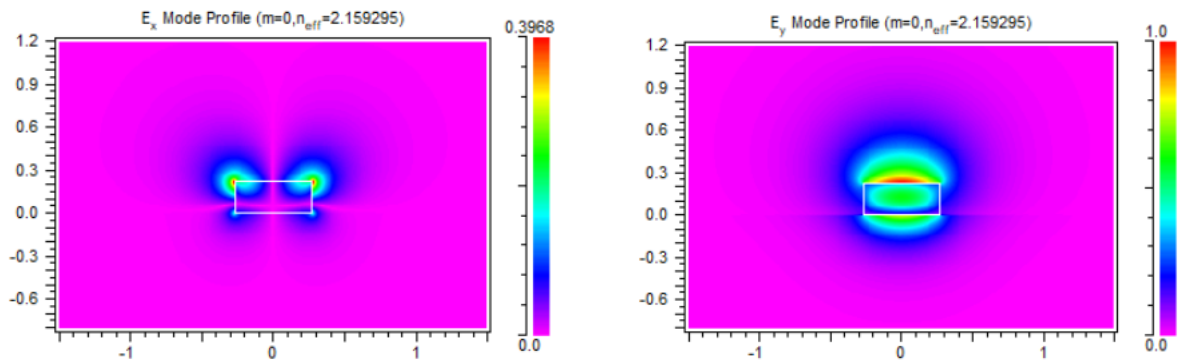


Figure 3. Calculated TM full-vector mode

BPM Simulations

The structure is set up in RSoft CAD with two pathways and two overlap monitors for both ports, as shown in Figure 4.

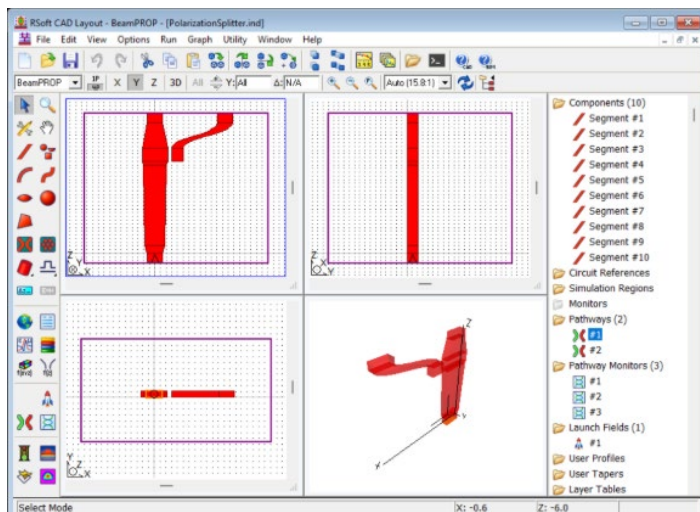


Figure 4. Layout in RSoft CAD

Since the final desired result is TE polarization regardless of the input polarization state, both pathway monitors are overlapped with the TE mode shown in Figure 2. Note the overlapping results are valid only at two ends of the device. The intermediate values are for reference only because the waveguide width changes continuously along the transition. In addition, a total power monitor is also set to count the total insertion loss. More pathway monitors, such as TM mode overlaps, can be set as well with extra computation effort during the propagation.

The transverse mesh size and the longitudinal propagation step size are each set to 10 nm. With such fine mesh, full-vector BPM with the standard Crank-Nicolson scheme (scheme parameter of 0.5) shows some noise accumulation and instability at the end as shown in Figure 5.

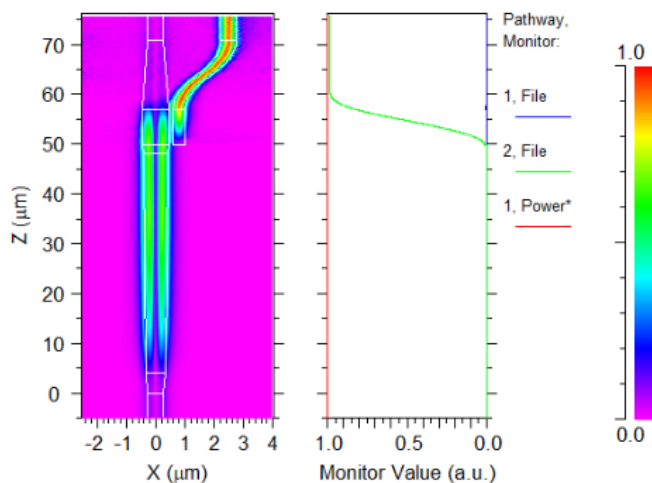


Figure 5. Full-vector BPM simulation for TM input with Crank-Nicolson scheme

With a slightly larger parameter value of 0.51, we are able to suppress the noise and obtain stable results. By applying the power-recalibration technique [3], we achieve stable results for both TE and TM inputs while conserving total power, as shown in Figure 6.

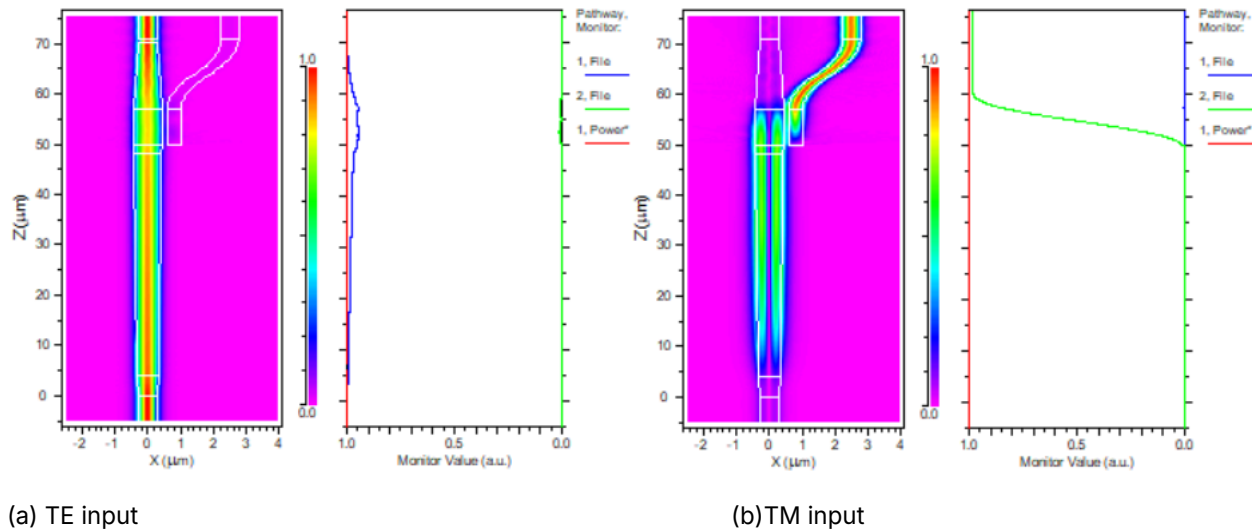


Figure 6. Auto-calibrated full-vector BPM simulation results with scheme parameter of 0.51.

The monitored outputs at the two ports are **0.9983** and **0.9818** for TE and TM inputs.

The simulation takes **0.07GB RAM** and **4.13 minutes** on a 32-core computer with an Intel CPU of Xeon Gold 6448Y 2.1GHz.

FDTD Benchmark

To validate BPM results, we made a benchmark with rigorous FullWAVE FDTD simulation on the same computer using the same number of cores (32). The FDTD simulation took **1200 minutes** and used **162GB RAM**. In comparison, BPM is **290x** faster and saves **2300x RAM** than FDTD.

The FDTD results are **0.9967** and **0.9745** for TE and TM inputs, which are nearly identical to the BPM results of **0.9983** and **0.9818**. The differences — **0.16%** and **0.75%** — are negligible compared with typical manufacturing variations.

It is well known that GPUs can dramatically accelerate FDTD simulation. FullWAVE FDTD has been implemented on GPUs, achieving up to a 80x speed-up on four H100 GPUs [5]. We also benchmarked the polarization-converter example on the same GPU machine. The test shows a runtime of **39.2 minutes** and a GPU RAM memory usage of **112GB**, in addition to **65GB of CPU RAM** for data processing. Compared with a 32-core CPU, the 4-H100 GPU is about **30x** faster — less than expected due to the substantial CPU workload required for monitor-data processing, even display was turned off completely. As expected, FDTD simulations on GPU and CPU give exactly the same results.

Therefore, BPM is on a computer with 32-core CPU is still **10x** faster than FDTD on a computer with 4-H100 GPU.

As a time-domain simulator, FDTD can perform pulsed simulation to obtain the whole spectrum with a single simulation at the cost of more computation resources. For this structure with 10 nm mesh size, pulsed FDTD simulation took **5208 minutes (3.6 days!)** and used **292.6G RAM** on CPU. Compared with CW simulation, Pulsed simulation is **4.3x** more in computation time and **1.8x** more in RAM.

The most important drawback for pulsed FDTD simulation is that mode dispersion is ignored, because only one modal profile at the central wavelength can be launched and used for overlapping outputs.

Shown in Figure 7 are spectral responses for TE and TM inputs by FDTD pulsed simulation. For the sake of comparison, we also performed wavelength scan in BPM. The results are shown in the same charts. As observed, BPM results are very close FDTD results over the entire wavelength range tested.

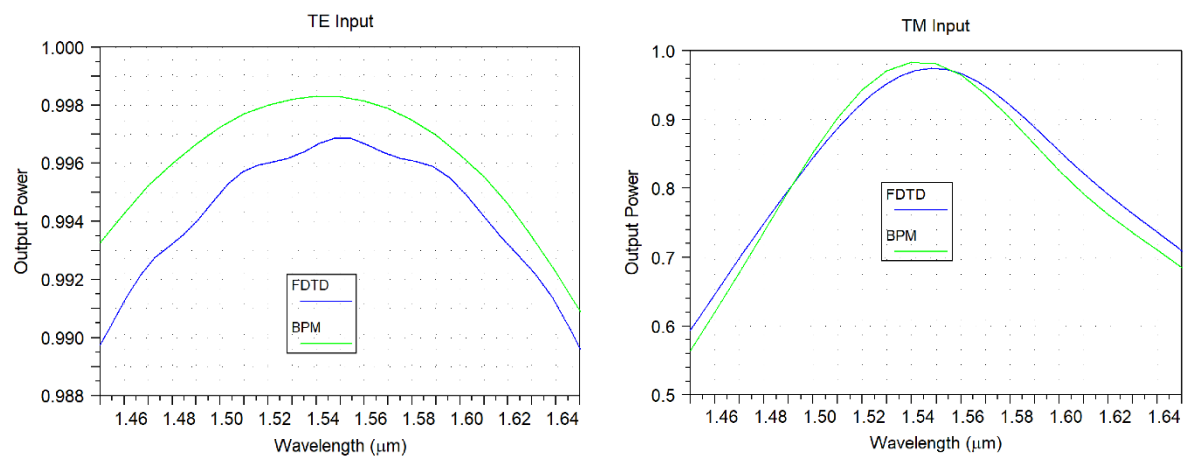


Figure 7. Spectral responses by both FDTD and BPM.

BPM still consumes 0.07 GB of RAM since the wavelength sweep runs sequentially, one wavelength at a time. Hence, the computation time is linearly proportional to the number of wavelengths, i.e., 86.7 minutes for 21 points in the range of 1.45 μm-1.65 μm.

MOST Clustering

BeamPROP BPM utilizes multi-cores via multi-threading by dividing a single job. Based on Amdahl's law, multi-threading efficiency $S(N)$ cannot be linearly proportional to the core number N , because only P percent of the code can be parallelized. Figure 8(a) shows the behavior predicted by Amdahl's law.

$$S(N) = \frac{1}{(1 - P) + P/N}$$

As noted, multi-threading efficiency is strongly dependent on the level of code parallelization. In this example, the efficiency in Figure 8(b) is reduced because the monitor calculations run exclusively on the master node and cannot be parallelized.

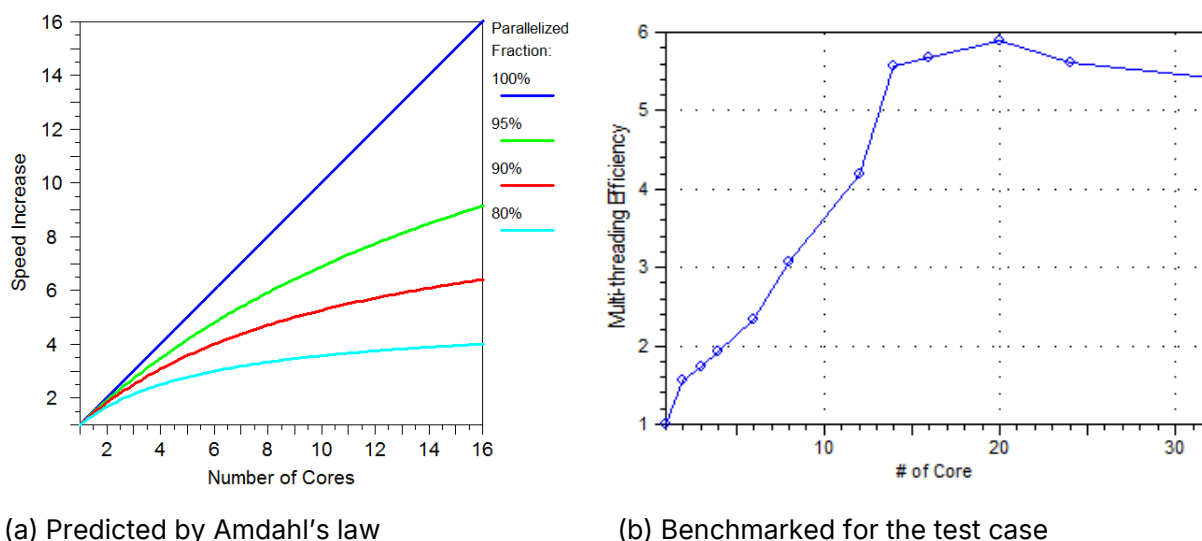


Figure 8. Multi-threading efficiency for the chosen example.

To make more efficient use of computing resources, multiple jobs can be launched simultaneously, since each BPM simulation requires very little RAM. The built-in Multi-variable Optimization and Scan Tool (MOST), can distribute and manage those scan jobs automatically. As shown in Figure 9, eight jobs run simultaneously and each job uses four cores to keep the total core numbers as 32 for a fair comparison.



Figure 9. MOST Clustering settings.

With the MOST Clustering setting, we can reduce the total simulation time from 86.7 minutes to 26.1 minutes. More savings can be achieved by optimizing the clustering settings.

To make the comparison self-consistent with FDTD, a fixed modal profile at wavelength 1.55 μm is used during the BPM scan, though dynamic mode for both launch input and overlap output can be easily used in BeamPROP.

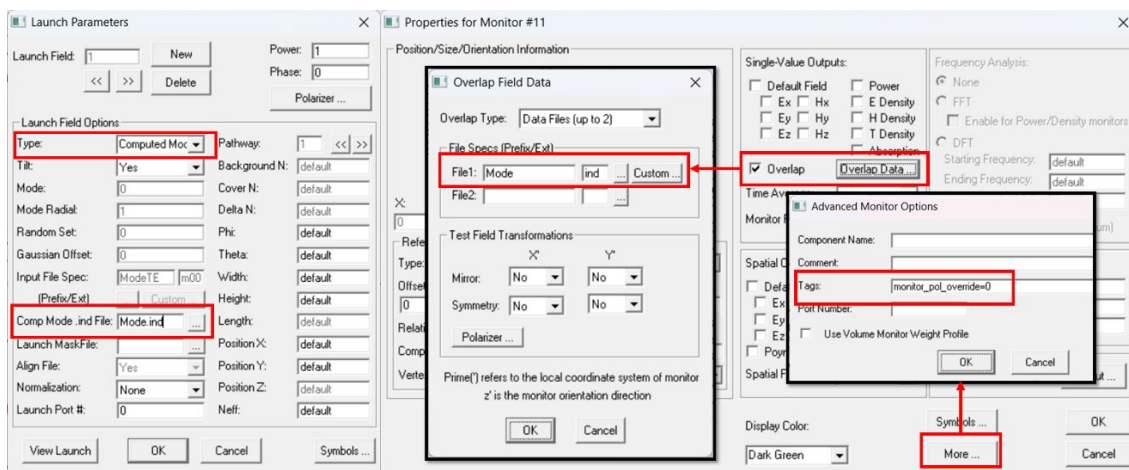


Figure 10. Dynamic mode calculations for input and output.

To obtain more accurate BPM results, dynamic mode calculations are used for both input and output overlap during the BPM wavelength scan, as shown in Figure 10. In addition to specifying the IND file for mode calculation, the polarization of the output overlapping monitors must be overridden using a Tag with the key "monitor_pol_override", because the monitored output is always TE mode — even when the input is TM.

The test results are shown in Figure 11, together with fixed mode for the sake of comparison. As observed, mode dispersion affects the BPM result only ~0.6% for TE input, but ~5% for TM input within the test wavelength range. Please note that the chart maybe deceiving, visually because of the plotting range in vertical axis.

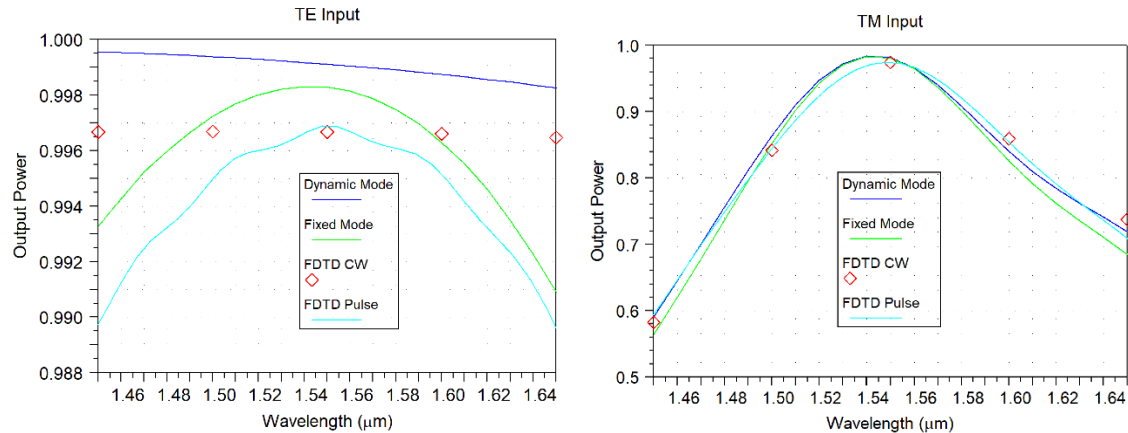


Figure 11. Comparison of BPM results with fixed mode and dynamic mode calculations.

Because excessive computer resources are required, the CW FDTD simulation was done at five wavelengths only and shown in the same chart together with Pulsed FDTD results for the sake of clear comparison. It is confirmed that mode dispersion also affects the FDTD results, 0.7% for TE and 3.8% for TM at wavelength 1.65 μm . Hence, BPM wavelength scan with dynamic mode calculation is more accurate than FDTD pulsed simulation for wide bandwidth.

Conclusion

In summary, BPM can be used to simulate photonics devices efficiently and accurately, including silicon photonic devices with high index contrast, as long as the key assumption — negligible backward reflection — remains valid. Compared with FDTD, BPM can reduce computation time and memory consumption by hundreds to thousands while producing results of comparable accuracy. Although high-performance GPUs can significantly accelerate FDTD, BPM remains substantially faster.

References

1. Dai, Daoxin, and John E. Bowers. "Novel concept for ultracompact polarization splitter-rotator based on silicon nanowires." *Optics express* 19.11 (2011): 10940-10949.
2. Huang, W. P., and C. L. Xu. "Simulation of three-dimensional optical waveguides by a full-vector beam propagation method." *IEEE journal of quantum electronics* 29.10 (2002): 2639-2649.
3. Xu, Chenglin, et al. "Suitability of BPM simulation for silicon photonics." 2021 IEEE 17th International Conference on Group IV Photonics (GFP). IEEE, 2021.
4. Xu, C. L., W. P. Huang, and S. K. Chaudhuri. "Efficient and accurate vector mode calculations by beam propagation method." *Journal of lightwave technology* 11, no. 7 (1993): 1209-1215.
5. [Adding GPU Acceleration to Electromagnetic Simulation](#), Keysight Optical Design Engineering Blog

Keysight enables innovators to push the boundaries of engineering by quickly solving design, emulation, and test challenges to create the best product experiences. Start your innovation journey at www.keysight.com.