FREE-ELECTRON—LASER SIMULATIONS ON THE MPP

Scott A. Von Laven*, KMS Fusion, Inc., Ann Arbor, Michigan and
Lorie M. Liebrock, Michigan Technological University, Houghton, Michigan

ABSTRACT

Free-electron lasers (FELS) are of interest because they provide high power, high efficiency, and broad tunability. FEL simulations can make efficient use of computers of the MPP class because most of the processing consists of applying a simple equation to a set of identical particles. A test version of the KMS Fusion FEL simulation, which resides mainly in the MPPs host computer and only partially in the MPP, has run successfully.

KEY WORDS: laser simulation, free-electron laser, massively parallel processing

INTRODUCTION

Free-electron lasers (FELS) have demonstrated high power output, high efficiency, and broad tunability from the microwave to the visible spectrum [1]. One-dimensional analyses, in large part, have guided the development of FELS to this point. As experimenters strive to optimize performance, the importance of more detailed analyses is increasing.

An FEL produces radiation when a relativistic beam of electrons passes through a periodic static transverse magnetic field (the wiggler shown in Fig. 1). The electron trajectories are perturbed by the radiation, and the beam becomes bunched on the scale of the radiation wavelength. This leads to coherence and high power. Gain, power at saturation, coherence, bandwidth, and other aspects of laser performance depend on the details of the interaction between the electrons and the fields.

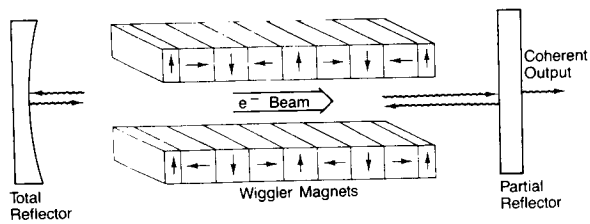

Figure 1. Conceptual FEL

SIMULATION

We simulate the operation of an FEL by following the three dimensional trajectories of a beam of electrons through a wiggler. The inhomogeneous wave equation for a particular mode is used to update the amplitude and phase of that mode; the radiation produced at a number of discrete frequency channels is calculated to determine the gain as a function of laser frequency. The Lorentz force equation is used to update the positions and velocities of the simulation electrons. Updating particle information, according to this simple prescription, accounts for most of the processing time and seems to be a task well suited to the MPP. For serial simulations on the KMS VAX 11/750, we restrict the number of particles to the order of 100. The MPP can accommodate many more particles. Consequently we can study a wider range of beam density

*Now at Mission Research Corp., Albuquerque, NM.

profiles and energy distributions. On the 11/750 we restrict the radiation to a single mode and a few channels. On the MPP we can use more channels and more modes.

In principle one can structure the simulation so as to use a small amount of memory per processing element (PE) and to distribute the computational load evenly across PEs. For the present form of the simulation this is best done by assigning one particle to each PE. The particle information is stored in the local PE memory and is updated at each time step by the PE.

In another possible mapping to the MPP, we could assign each PE a unique electron-channel pair. In this case each 32-by-32 subarray, whose southeast corner PE is directly readable by the master control unit (MCU) of the MPP, represents one frequency channel. Each PE within the subarray is responsible for the spatial coordinates and velocity components of one electron. The same set of 1024 electrons (or multiples of that number) is mapped onto each subarray. The radiative contributions of the electrons to a particular channel are computed and summed within the channel's subarray. The new channel amplitudes and phases are read from the respective corner PEs by the MCU, which can effectively broadcast to all PEs the field information necessary to update the electrons' velocities and positions. More electrons, channels, and modes can be handled by partitioning the simulation into 128-by-128 pieces each with a structure similar to the mappings described above.

As a development strategy we elected to make heavy use of the host-to-MPP call capability. Routines were converted from Fortran to MPP Pascal one at a time starting at the lowest level routines. In this way errors generated during code conversion were more easily isolated and corrected. A disadvantage of this method is that the routine and stager calls must be carefully rewritten for each intermediate version. A simpler software interface between the MPP and its host, a friendlier debugger, and more PE and MCU memory would reduce this disadvantage and permit code development for all users to proceed more smoothly and more quickly.

The structure of the code is shown in Fig. 2. The diagnostics and field pushing routines are called much less frequently than the particle pushing routine (inside circle). The latter currently run in the MPP while the rest of the code runs in the host. A description of the physical model was published earlier (ref. 2).

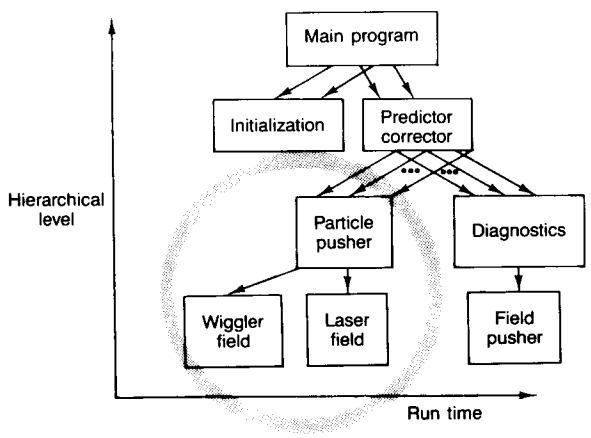

Figure 2. Structure of KMS Fusion 3D FEL simulation

RESULTS

Direct comparisons between serial runs and parallel runs with 16,384 particles have not been performed. Instead, fewer particles were used and the results interpreted accordingly. Until all loops over the particles are in the MPP, even the parallel runs must be performed

with a restricted number of particles. Overall processing and elapsed time are the only performance diagnostics used so far. Thus these results are qualitative. For serial runs processing time increases slightly faster than the number of particles. A typical run with 100 particles traversing a one-meter interaction region requires a few minutes of processing time on the host VAX 11/780. Of the runs involving the MPP, those employing the code version represented by Fig. 3 are of the greatest interest. For these runs (with 100 particle or fewer) the processing time is a few minutes and does not increase much with the number of particles. This is because fewer of the loops over the particle index are being executed serially. Versions with only one or two MPP-resident routines required three to five times more processing time.

Only the actual application of the incremental change in particle phase-space coordinates (performed by the predictor-corrector) remains to be parallelized. Then 16,384-particle one-meter runs should still only require a few minutes of processing time. An attempted parallel implementation of the predictor corrector required the use of the stager for parallel-array storage. Ironically, the extra statements required for data swaps pushed the size of our code beyond the capacity of the MCU memory. The generation of more efficient assembly code from MPP pascal may solve this problem. Otherwise we may need to switch to a simpler predictor corrector at the expense of imposing a smaller time step.

ACKNOWLEDGEMENT

Tim Busse provided expert programming assistance throughout this effort.

REFERENCES

1. T.C. Marshall, "Free-Electron Lasers," Macmillan, New York, 1985.

2. H. Takeda, S. Segall, P. Diament, and A. Luccio, "Stable Off-Axis Electron Orbits and Their Radiation Spectrum in a Helical Wiggler," Nucl. Instr. and Methods, **A237**, 234 (1985).