Particle Orbit Tracking on a Parallel Computer: Hypertrack

B. Cole, G. Bourianoff, F. Pilat
SSC Laboratory*
2550 Beckleymeade Avenue
Dallas, Texas 75237

R. Talman
Cornell University
Ithaca, New York 14853

Abstract

A program has been written which performs particle orbit tracking on the Intel iPSC/860 distributed memory parallel computer. The tracking is performed using a thin element (TEAPOT-style[1]) approach. A brief description of the structure and performance of the code is presented, along with applications of the code to the analysis of accelerator lattices for the SSC. The concept of "ensemble tracking", i.e. the tracking of ensemble averages of noninteracting particles, such as the emittance, is presented. Preliminary results of such studies will be presented.

Introduction

Accurately tracking the orbits of a group of particles in a synchrotron for sufficient duration to simulate the normal operation of such a machine is a computationally demanding task. The first improvement over ordinary scalar computation was realized when tracking codes were modified to operate on vector computers such as the Crays [2]. However, the high cost of computer time on such machines make it infeasible to use them for operational simulations.

Much of the particle dynamics involved is single particle dynamics. Thus, the particles can be treated independently, and the process can be mapped efficiently onto a parallel architecture, provided the individual processors have both sufficient processing power to handle tracking the individual particles effectively and fast access to the data describing the accelerator. In fact, the Intel Hypercube is such a machine, since it has i860 RISC architecture individual processors, each of which with 8Mb. memory.

Interface

Before performing tracking studies on a supercomputer, preliminary work is required. From the basic design, errors must be included, a closed orbit found, and compensation performed Only after these are done is the lattice description ready for the longer term studies. During the development of the vectorized tracking code ZTRACK, it was realized that this could be most efficiently done by performing the initial operations on local workstations, then transferring a file containing the complete description of the lattice, with errors and correction in place, to the Cray. This eliminated the need for porting complex tuning algorithms to the supercomputer. The same strategy is used in Hypertrack; in fact, the same file format has been maintained.

Code Structure and Usage

The code is based on a manager/worker model as shown in Figure 1. The single manager node first reads in the lattice data and command instructions from the input file, then parcels out tasks (particles to be tracked) to the workers, and waits for the workers to finish. If there are more tasks then there are workers, the manager continues assigning until all jobs have been started. Once the manager is notified that the last worker is finished, it terminates the program. This complexity is hidden from the user, however, by a server program running on the System Resource Manager. This program performs the system-level tasks involved.

Physical Content of the Code

Hypertrack uses a thin element method for tracking

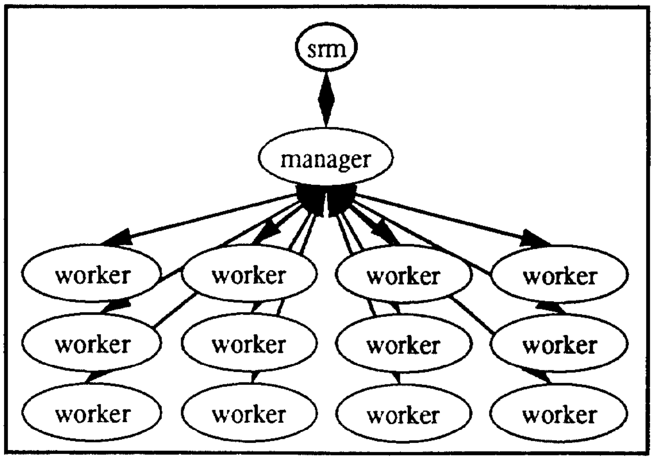

Fig. 1: Schematic of Hypertrack program structure. The IPSC nodes are shown as ovals.

---

*Operated by the Universities Research Association Inc., for the U.S. Department of Energy under Contract DE-AC02-89ER40486.

particles, in which the passage of the particle through each thin element is modeled accurately, including the difference in path length for off-momentum. If there are RF cavities in the lattice description file, the user is allowed to apply RF to the system, either in coasting beam mode or to simulate acceleration through a portion of a cycle. In the current version of the code, the user must specify a profile for the momentum of the reference particle, a profile for the maximum accelerating kick per turn (MeV), the harmonic number h, and the value of the momentum compaction α. The code calculates the reference phase as each particle passes through an RF cavity by solving the simple relation

$$\phi_s = \operatorname{asin}\left((\Delta E) / (eV) \right)$$

The code then compares the reference energy with the transition energy (if it exists; α is permitted to be negative) to reference the phase correctly.

Performance & Initial Results

Each i860 in the Hypercube is a RISC architecture processor running at 40 MHz. peak rated at 60 MFlops (double precision). Thus, the total capability of the hypercube at the SSC is 3.8 GFlops. If one considers that each node on the Hypercube has the same order of magnitude of processing capability as a desktop workstation, then one can model the Hypercube as 64 closely connected workstations.

A more pertinent performance measurement is the time required for a single node to track a particle through an element in an accelerator, from which one can calculate the time required to track a single particle through a given lattice. From this we arrive at the following estimate of time for a given run:

$$t = 1.25\times10^{-4} N_{elem} N_{turns} \operatorname{int}\left(\frac{n_{particles}}{n_{nodes}} + 0.49 \right)$$

For smaller machines such as the LEB with ~1000 elements, the ability to model exactly the longitudinal and transverse dynamics while changing the reference momentum provides one with the ability to model the behavior of a particle through the acceleration cycle. Figures 2 and 3 show the results of such a run for the LEB, with the initial and final phase space plots for 512 particles with a gaussian distribution tracked through the entire acceleration cycle (24000 turns).

For a large machine such as the collider, with 2.4×10^4 elements, we see that particles will be tracked at ~2.5 sec./turn, making this paradigm ill-suited for long-term tracking of large machines. However, from the point of view of operational simulation many interesting effects manifest on the order of 1000 turns, which can be handled reasonably. Figure 3 is a dynamic aperture result based on relatively short-term (1000 turns) tracking. A grid of 625 particles evenly distributed in x and y are tracked through the machine and the number of turns survived recorded. Also, the code records the element number in which the particle was lost, thus allowing for beam loss studies. Figures 4 and 5 depict results from a recent run on the collider lattice. In Figure 4, it can be seen that the distribution is offset from the center of the beamline due to the closed orbit. Comparing Figure 5 with a lattice description, one can identify the locations of particle loss, which for this simulation appears to occur in the arcs.

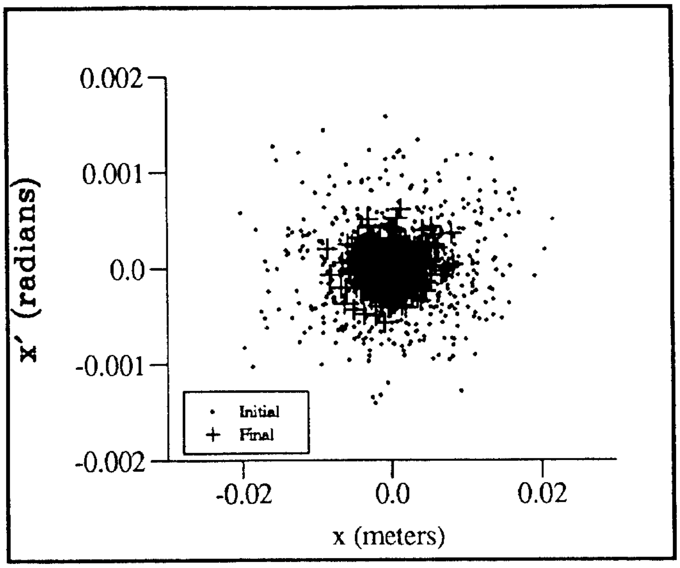

Figure 2: Horizontal Phase Space for the full LEB cycle

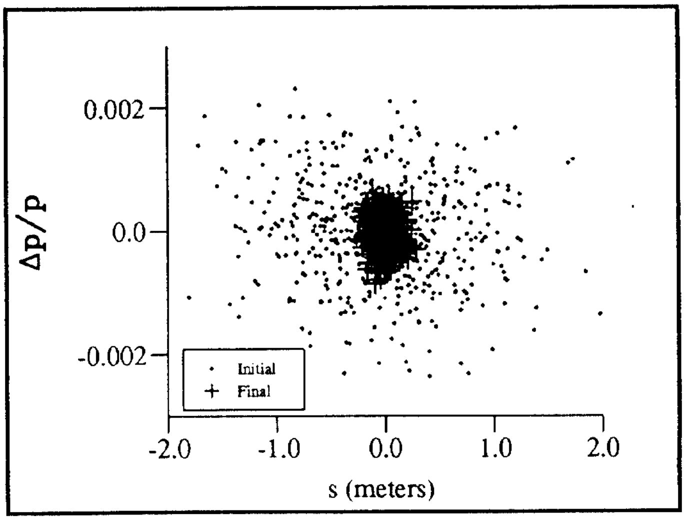

Figure 3: Longitudinal Phase Space for the full LEB cycle.

Ensemble Tracking

In a real accelerator, the dynamics of a single particle is rarely (if ever) measured: measurements are made on bunches. Simulation results, however, have focussed upon obtaining information about statistical distributions by examining the behavior of small numbers of test particles. The ability to track an ensemble (100 to 10000 particles) through the machine simultaneously, hence the name "ensemble tracking," allows us to model the behavior of a bunch as a whole.

A preliminary attempt at ensemble tracking may be obtained by post-processing the data generated during the LEB

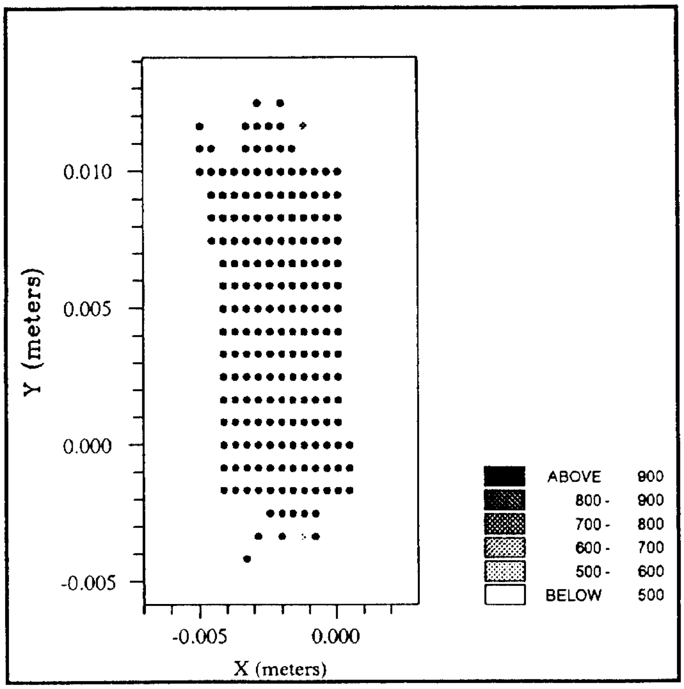

Figure 4: Survival plot for the current collider lattice. The entire region was populated with a grid of particles. The dark area contains particles still circulating after 1000 turns.

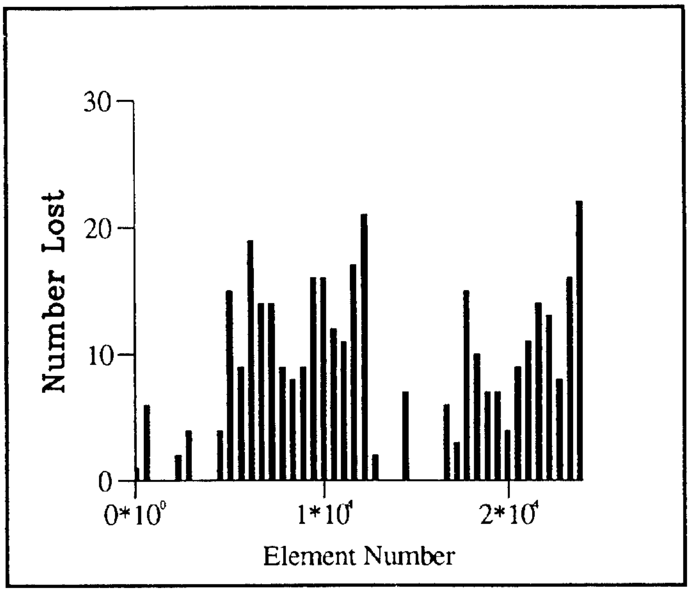

Figure 5 Histogram of the number of particles lost in each element. Particles which were lost in the first turn are omitted.

run previously discussed. During the run, the full phase space coordinates of each particle are recorded every 500 turns for the full run. The data in these files are then sorted by turn number, generating "snapshots" of the ensemble at a number of points in the cycle (48). Then for each snapshot, moments in all phase space coordinates are obtained. From this information and the values for Courant-Snyder parameters, we calculate the transverse emittances in both planes. Figure 6 is a plot of these quantities, along with s_{rms}.

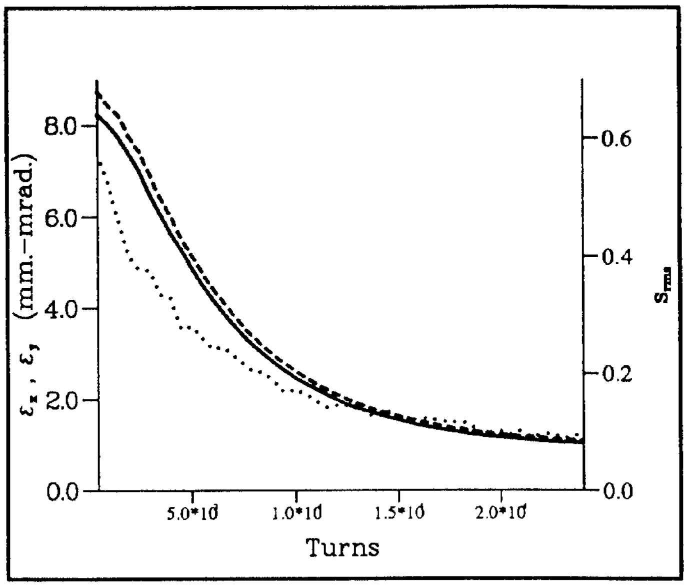

Figure 6: Profiles of transverse emittances (solid and dashed) and s_{rms} through the LEB cycle.

Conclusions and Future Work

A new particle tracking code utilizing the capabilities of the Intel iPSC/860 Hypercube has been developed. This code accurately models the single-particle dynamics of protons in synchrotrons. Further, this code models both the longitudinal and transverse dynamics of particles undergoing acceleration in a synchrotron, using user-specified profiles for the cavity voltage and reference momentum. Because it tracks a number (64) of particles simultaneously, it is well suited for studies involving the behavior of large numbers of particles. In particular, for the LEB, the code can model the behavior of a collection of 512 particles during an entire cycle. Ensemble tracking, the measurement of the evolution of statistical properties of beams based upon tracking results from a statistically large number of particles, provides a new tool for the study of beam dynamics.

Current work in progress is concerned with the generalization of the formalism to allow arbitrary phase specification for rf cavities and multiple cavities with individual cavity profiles. In the future, Hypertrack will undergo alteration to increase the number of particles that can be simultaneously tracked to provide better statistics and provide a starting point for modelling collective effects for the LEB at injection and the MEB during transition. The inclusion of time-dependent variations in the multipole fields to model tune modulation will also be undertaken.

References

[1] L. Schachinger and R. Talman, "TEAPOT: A Thin-Element Accelerator Program for Optics and Tracking," Particle Accelerators 22, 35 (1987)

[2] Y. Yan, G. Bourianoff, and L. Schachinger, "A Typical Longterm Tracking Results for SSC Aperture Studies," SSCL-303, unpublished.