# Problems

3. Write a parallel version, PLLSQR, of the linear least squares routine of Figure 2.2.2. You can follow closely the pattern used by PLINEQ (Figure 6.2.1), and distribute the columns of $A$ cyclically over the available processors. In REDQ, whoever has the active column L should broadcast it to the other processors, who should save it in a vector COLUMNL, since all need access to this column. Each processor will need to modify COLUMNL itself, as well as their own columns, since COLUMNL(I) changes and is used, each pass through the loop 10. The back substitution can be done almost exactly as in PLINEQ. ERRLIM can be passed as an argument to PLLSQR, if desired.

   Since PLLSQR should find the exact solution of $A\boldsymbol{x} = \boldsymbol{b}$ if there is one, you can test your routine on the system 1.9.10 again, simply repeat both parts of Problem 2 using PLLSQR in place of PLINEQ. Physically distributing the columns of $A$ over the processors (part (b)) will be easy, as in PLINEQ. However, switching the order of the REDQ loops 5 and 10, to produce a unit stride in the inner loop, would be more difficult. Why?

6. Repeat Problem 3a, only this time use a parallelized version, PREDH, of REDH (Figure 2.3.1) instead of REDQ. Again, distribute the columns of $A$ cyclically over the available processors. In PREDH, whoever has the active column $L$ should call CALW to compute the vector $W$, and broadcast $W$ to the other processors. $A(I, L)$ will also need to be broadcast to the other processors after it changes.