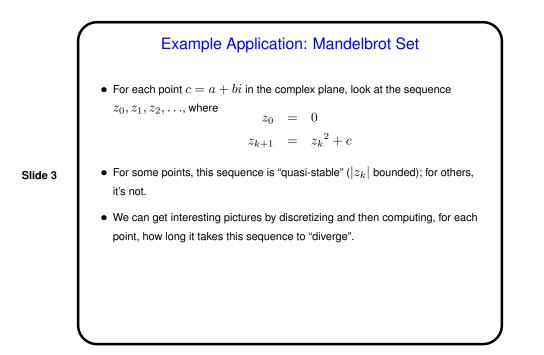
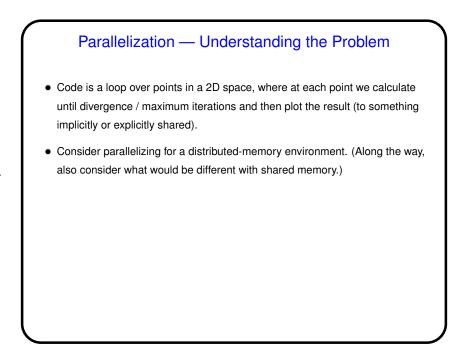


Slide 1

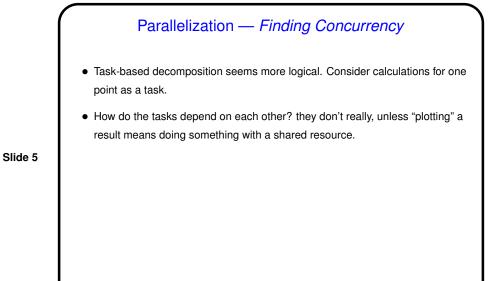
Master/Worker Example, Continued Last time we looked at code for a generic master/worker program, in which "tasks" just consist of sleeping a specified number of milliseconds. Testing uncovered a bug — code to sleep didn't work for times over 1 second. Problem is that I was not using this function correctly. See revision.

Slide 2





Slide 4



Slide 6

Parallelization — Algorithm Structure • Many mostly-independent tasks, forming a flat set rather than a hierarchy, so Task Parallelism seems like a good choice. • Key design decisions are how to assign tasks to UEs, how to manage "plotting". • Probably makes sense to group tasks by rows rather than individual points. We could try a simple static distribution, but might have to do something more complex if that doesn't give good load balance. • Managing plotting? in a distributed-memory environment, might make sense to just assign that job to a process that does nothing else.

