Parallelization of Heart Practical 6

Joe Pitt-Francis

May 4, 2005

1 Introduction

This document concerns the parallization of heart practical 6. As of this date the code which runs the practical is contained in `Chaste/heart/tests/TestMonodomainDg0Assembler.hpp` and consists of tests of the form `testMonodomainDg0?D` (where ? is replaced by the dimension of the problem). All these tests have the same basic structure:

1. A mesh is constructed from a mesh reader.
2. An `EulerIvpOdeSolver` is constructed.
3. A `MonodomainPde` object is constructed using the number of nodes in the mesh, the Euler solver and some time parameters. This object will contain the ODE parameters, ODE solutions and stimulus functions at each of the node locations.
4. The initial conditions at each location are set by preparing a std::vector and using `SetUniversalInitialConditions`.
5. A stimulus function is set on (at least) one of the nodes.
6. A boundary condition container is constructed to contain Neumann boundary at all boundaries.
7. A `SimpleLinearSolver` (which is a wrapper to a KSP) is constructed.
8. A `MonoDomainDg0Assemble` object is constructed.
9. A vector of voltages (-84.5 for each node) is built.
10. A data writer is initialized.
11. In a loop over time
    - The voltages are used to set the initial condition of the assembler.
    - The assembler is used to solve the PDE (by repeatedly assembling and solving a linear system). This returns a new vector of voltages which is used in the next iteration.
    - The data writers write some data.
12. The final solution may be tested against some known solution.

2 What can by parallelized?

2.1 Linear algebra

The linear algebra components of the code are already written in parallel since they use PETSc vectors and matrices. The core linear algebra solver (KSP) is aware of the parallel structure of the matrices and vectors and is written in parallel. There are many efficiency savings to be made in terms of sparse matrix storage and packing the matrix block diagonal, but these won't effect the correctness of the solver.

A potential problem is in interrogating PETSc vectors in order to test solutions when the code is run in parallel. Consider this code (which is correct when run sequentially:

```
PetscScalar *solution_elements;
VecGetArray(solution_vector, &solution_elements);
TS_ASSERT_DELTA(solution_elements[0], 1.0, 0.000001);
TS_ASSERT_DELTA(solution_elements[1], 2.0, 0.000001);
TS_ASSERT_DELTA(solution_elements[2], 3.0, 0.000001);
```

The problem with this test is that `solution_vector` will be split between processors. When run on two processors, process 0 owns the first 2 components of the vector and process 1 owns the final component. The pointer `solution_elements` points to the portion of the vector which is locally owned. If the above code is run on two processors, process 0 will fail the final assertion (since it doesn't own the final element), and process 1 will fail all the assertions. The correct code is to only tests the locally owned portions of the vector:

```
PetscScalar *solution_elements;
VecGetArray(solution_vector, &solution_elements);
int lo, hi;
VecGetOwnershipRange(solution_vector,&lo,&hi);
double real_solution[3]={1.0,2.0,3.0};
for (int i=0;i<3;i++){
  if (lo<=i && i<hi){
    TS_ASSERT_DELTA(solution_elements[i-lo], real_solution[i], 0.000001);
  }
}R
```

2.2 Linear system assembly

Three solutions...

2.3 ODE solvers

3 Fine-grained tasks

3 Make separate directory for parallel tests which are always run on more than one virtual processor. (Done Tues AM).

1 Fix some linear algebra tests so that they pass when run in parallel. (Done Tues AM).

3 Fix linear system class so that assemblers can only add or insert to values that are held locally. (Done Weds AM).

3 Make a simple PDE on a mesh work (Done Tues PM for heat equation on a disk).

15 Make `MonodomainPDE` into a distributed object. The distribution should mirror the distribution of a PETSc vector.

15 Make data writers work in parallel.

2 Copy and fix the three Heart Practical 6 tests.

42 (Total estimate in pair hours. 32 remaining)