

---

CIS 510 FINAL PROJECT

DISTRIBUTED PARTIAL DIFFERENTIAL  
EQUATION SOLVER FOR  
3-DIMENSIONAL MODELS  
(DIFFYQ)

---

**Project Team:**

Adam Martini

Wes Erickson

Ran Tian

April 21, 2014

# Executive Summary

We will design a flexible system for solving large partial differential equations in a parallel distributed system. The product of our this project will solve 3-dimensional models of any size for a given number of time steps. Our system will take any size model and process it efficiently in parallel on multiple nodes.

## 1 Project Description

Diffyq is a distributed partial differential equation (PDE) solver for 3-dimensional models. This systems takes advantage of the memory and processing power of multiple nodes to efficiently compute the PDE. Given a 3D dataset, a user defined set of update equations, a time step granularity, and a number of steps; our system automatically marshalls the data and computational code to multiple nodes to initialize processing. The PDE is then computed for the given number of steps and the final result of stored on the main node.

A primary goal of this project to create a generalized application that can used to solve the PDE for any waveform. To accomplish this, we allow the user to provide a set of user defined functions as input for computation. Our system uses these functions to iteratively update the PDE matrix during parallel processing. We carefully outline the format for these user defined functions in Section 2.

Communication between nodes allows the system to calculate the border values of between the divisions of the PDE matrix. This is an example of the stenciling pattern for parallel processing. On each node, our program takes advantage of opportunities for shared-memory parallelism through map-reduce and divide conquer techniques for computation update components.

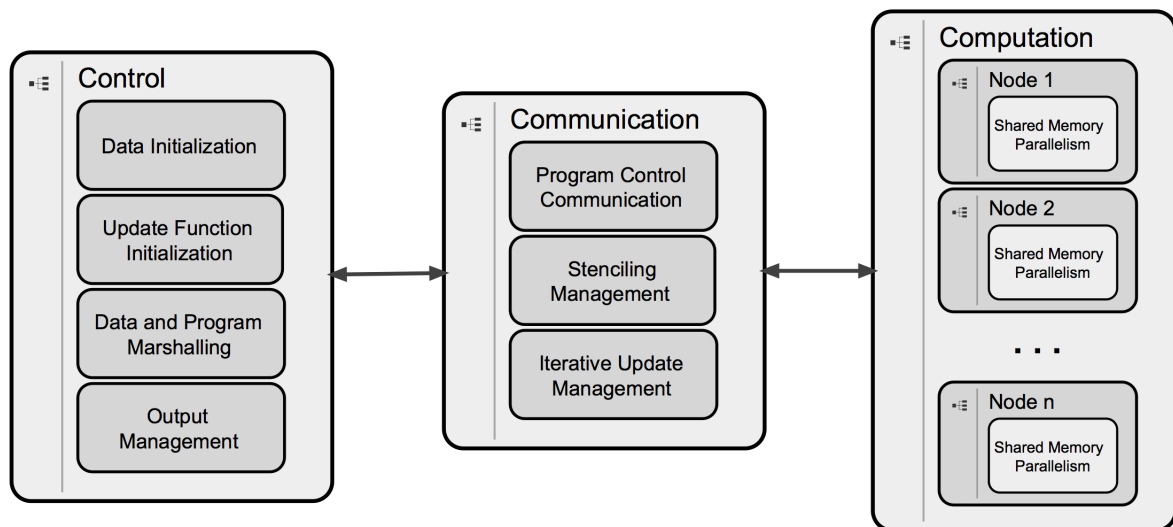
## 2 Highlevel Architecture

Figure 1 shows a block level diagram of the Diffyq software. There are three main components to the solution.

**Control Layer**

**Communication Layer**

**Computation Layer**



**Figure 1:** High Level Architecture Diagram

## 2.1 Control

## 2.2 Communication

## 2.3 Computation

# 3 Parallel Plan

The Diffyq design includes several places to introduce and explore/optimize parallelism.

**Data and Program Marshalling** Map Stenciling...

**Stenciling Management** Communication minimization along borders...

**Node Level Parallelism** Matrix Multiplication...

# 4 Project Schedule

Week	Deliverable
5	
6	
7	
8	
9	
10	