UNIVERSITY OF THE PHILIPPINES

Bachelor of Science in Applied Physics

Clarence Ioakim T. Sy

Learning dynamics in a cellular automata model of classroom peer-to-peer interactions

Thesis Adviser:

Johnrob Y. Bantang, Ph.D.

National Institute of Physics

University of the Philippines Diliman

Date of Submission:
June 2024

Thesis Classification:
F

National Institute of Physics
College of Science
University of the Philippines
Diliman, Quezon City

ENDORSEMENT

This is to certify that this thesis entitled **Learning dynamics in a cellular automata model of classroom peer-to-peer interactions**, prepared and submitted by Clarence Ioakim T. Sy in partial fulfillment of the requirements for the degree of Bachelor of Science in Applied Physics, is hereby accepted.

JOHNROB Y. BANTANG, Ph.D.
Thesis Adviser

The National Institute of Physics endorses acceptance of this thesis in partial fulfillment of the requirements for the degree of Bachelor of Science in Applied Physics.

WILSON O. GARCIA, Ph.D.
Director
National Institute of Physics

ABSTRACT

LEARNING DYNAMICS IN A CELLULAR AUTOMATA MODEL OF CLASSROOM PEER-TO-PEER INTERACTIONS

Clarence Ioakim T. Sy
University of the Philippines (2024)

Adviser:
Johnrob Y. Bantang, Ph.D.

Peer instruction has recently become one of the popular means of classroom instruction in Physics Education. Such educational setup must involve both physical interaction with things and actually doing some procedural steps mentally or physically. In this study, we investigate the effects of different seating arrangements on the students' learning efficiency in peer instruction by modeling the transfer of knowledge within the class as a probabilistic cellular automata model. We compared the efficiency of learning between the traditional learning model and the peer instruction model. We found that in square classrooms with different lengths $L \in \{32, 48, 64, 96, 128\}$, the inner corner seating arrangement performed the best among the peer instructions setups in terms of both the time t_{max} it takes for all the students to learn and the classroom's learning rate m. This result is different from a previous study, where they found that the outer corner seating arrangement performed the best. The difference stems from the simplifications made in this model that may not reflect real world factors. Our model uses binary values in an isotropic system and does not consider memory or unlearning. However, despite these simplifications, we found that in smaller classrooms with slow learners, peer instruction is more efficient compared to the traditional learning model, just as previous studies have suggested.

Taken from SPP-2024 paper

PACS: 01.20.+x [Communication forms and techniques (written, oral, electronic, etc.)], 01.30mm (Textbooks for graduates and researchers)

Table of Contents

List of Figures

Chapter 1

Peer Instruction and Traditional Models of Teaching

something

1.1 History of Peer Instruction

Something something Mazur something something

1.2 Difference of Peer Instruction from Traditional Models

1.2.1 Details of Peer Instruction

1.2.2 Benefits of Peer Instruction

1.2.3 Drawbacks of Peer Instruction

1.3 Numerical models for learning in the classroom

Chapter 2

The Classroom as a Binary Probabilistic Cellular Automata Model

The classroom is a complex system that can be modeled as a probabilistic cellular automata. This chapter will discuss the classroom as a complex system and the probabilistic cellular automata model. The chapter will also discuss the implications of the model on the classroom and the teaching-learning process. (AI Generated Text)

Probably add objectives here???

2.1 Cellular Automata and its use in modelling complex systems

A two-dimensional (2D) rectangular cellular automata can be defined by a five-tuple [1] + Reinier MS:

$$\text{CA} = \{\mathcal{S}, \mathcal{C}, \mathcal{L}, \mathcal{N}, \mathcal{R}\} \tag{2.1}$$

where

$\mathcal{S} =$ $\{0, \ldots, S-1\}$ is the set of representations of states each cell can take on.

$\mathcal{C} =$ $\{c = i, j \mid i \in \{1, 2, 3, \ldots, L_1\}, j \in \{1, 2, 3, \ldots, L_2\ \}\text{s.t. } L_1 \times L_2 = N\}$ is the set of identifiers for each cell in the automaton where N is the total number of cells and L_1 and L_2 are the lengths of each side of the automaton space. The cells can then be identified by their position in the automaton (i, j). So, the state of cell $c \in \mathcal{C}$ can be written as $s_c = s_{i,j} \in \mathcal{S}$

$\mathcal{L} =$ defines the lattice neighborhood which is generally a mapping $f : \mathcal{C} \to C^M$ where M is the number of neighbors of a cell $c \in \mathcal{C}$. Any given cell c is mapped to another tuple of cells: $L_{i,j} = \{(i-1, j-1), (i-1, j), (i-1, j+1), (i, j-1), \ldots, (i, j)$. Where r is the radius of the Moore neighborhood. We then say that $\mathcal{L}_{i,j}$ contains the set of neighboring cell for $c_{i,j}$.

$\mathcal{N} =$ $\mathcal{S}^M$, the set of neighborhood states. Thus, $N_c = N_{i,j} \in \mathcal{N}$ such that each $\mathcal{N}$ is in the form of the M-tuple $\{s_{i-1,j-1}, s_{i-1,j}, s_{i-1.j+1}, s_{i,j-1}, \ldots, s_{i,j}\}$.

$\mathcal{R} =$ defines the set of rules implemented in the CA with $g : s_{i,j} \mid \mathcal{L} \to \mathcal{S}$ as the mapping of any neighborhood state N_c to a new state $s'_{i,j}$ of the cell c. At the next time step, $s'_{i,j}$ replaces the original state $s_{i,j}$.

$\mathcal{N}$ can vary with the neighborhood structure and the boundary conditions of the automaton. The neighborhood structure dictates the shape the neighborhood in the lattice. Common neighborhood structures include the von Neumann (diamond) and Moore (square) neighborhoods. Boundary conditions dictate how the automaton treats cells at the egde of the lattice when determining the neighborhood. Common boundary conditions include toroidal, spherical, and fixed boundary conditions.

$\mathcal{R}$ can also be affected by other factors such as whether the rules are deterministic or probabilistic and whether they are implemented synchronously or asynchronously. An automaton with deterministic rules will always produce the same output given the same input, while an automaton with probabilistic rules will produce different outputs given the same input. In Conway's Game of Life, a cell dies when it has three live neighbors, while a cell is born when it has two or three live neighbors. This is an example of a deterministic rule. An example of a probabilistic rule would be a cell dying with a probability of 0.25 when it has three live neighbors. An automaton with synchronous rules will update all cells simultaneously, while an automaton with asynchronous rules will update cells one at a time. (something explanation something about sync vs async)

Due to the flexibility of cellular automata, they can be used to model a wide variety of complex systems. Cellular automata have been used to model physical systems such as fluid dynamics, biological systems such as the spread of diseases, and social systems such as traffic flow [3]. Its discreteness and locality make it a good model for systems that are composed of many interacting parts. Thus, we have chosen to use a two-state probability cellular automata to simulate the learning process for students in the classroom

2.2 PI as a Discrete Probabilistic CA Model

We used a two-dimensional binary probabilistic cellular automata (PCA) model to simulate the learning process in a classroom. In this PCA model, each cell in the automaton represents a student and the state of each cell represents their aptitude $S = \{\text{unlearned, learned}\} = \{0, 1\}$. We assign the neighborhood to be an outer-totalistic Moore neighborhood of radius $r = 1$ and define the boundary conditions to be fixed wherein the grid does not wrap around itself and $s_{i,j} = 0$ for $i, j \notin [1, L]$. The rules of the automaton describes how the students learn from their neighbors based on three parameters. First (1), the learning rate $\lambda_{i,j}$ of student $c_{i,j}$ which describes how receptive they are to peer instruction. Secondly (2), the relative spatial factor $\rho_{i+\delta i,j+\delta j}$ which describes how likely it is to learn from the neighbor $c_{i+\delta i,j+\delta j}$ based solely on their relative position with respect to $c_{i,j}$. Lastly (3), the aptitude level of the neighbor $s_{i+\delta i,j+\delta j}$ which dictates whether student $c_{i,j}$ can learn from them. The probability for a student to learn in each time step is then determined by the following equation:

$$P_{ij} = 1 - \prod_{\forall \delta i, \delta j} [1 - (\lambda_{ij})(\rho_{i+\delta i,j+\delta j})(s_{i+\delta i,j+\delta j})] \tag{2.2}$$

where

$P_{i,j} \in [0, 1]$ is the probability of $c_{i,j}$ to learn in each time step,

$\lambda_{i,j} \in [0, 1]$ is the learning rate of $c_{i,j}$ with values $\lambda_{i,j} = \{\lambda_0 \pm \delta\lambda\}$ where $\lambda_0 = 0.5$ and $\delta\lambda = \{0.1, 0.2, 0.3, 0.4\}$,

$\rho_{i+\delta i,j+\delta j} \in [0, 1]$ is the probability of $c_{i,j}$ to learn from their neighbors in seats $\{c_{i+\delta i,j+\delta j} \forall \delta i, \delta j \in \{-1, 0, 1\}\}$ solely based from their relative positions with each other, and

$s_{i+\delta i,j+\delta j} = \{\text{unlearned, learned}\} = \{0, 1\}$ are the neighbors aptitude level.

In the five-tuple form, the PCA model for the classroom can be written as:

$\mathcal{S} =$ {learned, unlearned} $= \{0, 1\}$

$\mathcal{C} = \{(1,1),(1,2),\ldots,(1,L),(2,1),(2,2),\ldots,(2,L),\ldots,(L,L)\}$ where L is the length of the square classroom.

$\mathcal{L} = f(c) \leftarrow [L_c = \{(i+\delta i, j+\delta j)\ \forall\ (\delta i \wedge \delta j), \delta i, \delta j \in \{-1,0,1\}\}$ as a mapping for outer-totalistic Moore neighborhood of radius $r = 1$ with a fixed boundary condition.

$\mathcal{N} = \{00000000, 00000001, \ldots, 11111111\}$ such that the representation of the neighborhood state $N_c \in \mathcal{N}$ is equivalent to $N_c = \{s_{i+\delta i, j+\delta j} \forall \delta i, \delta j \in \{-1,0,1\}\}$.

$\mathcal{R} =$ the probabilistic rule defined by equation 2.2.

The numerical procedure is outlined in Figure 2.1. Each simulation starts the classroom with four learned students $n_0 = 4$ placed in different seats in the classroom.

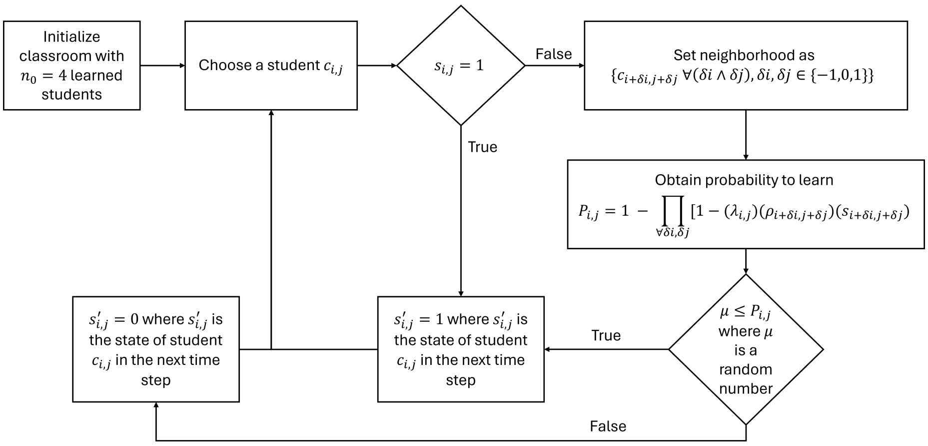

Figure 2.1: Numerical process for simulation of 2D BPCA for PI set-ups.

The seating arrangement (SA) were chosen from a previous study that showed that the SA can affect the learning process [4]. These SA's are namely: inner corner, outer corner, center, and random. The SA configurations are shown in Figure 2.2.

From the simulations, we compared both the average number time steps $\langle t_{max} \rangle$ it takes for all the students in the classroom to learn and the average learning rate $\langle m \rangle$ across different configurations over 5 independent runs. The learning rate for

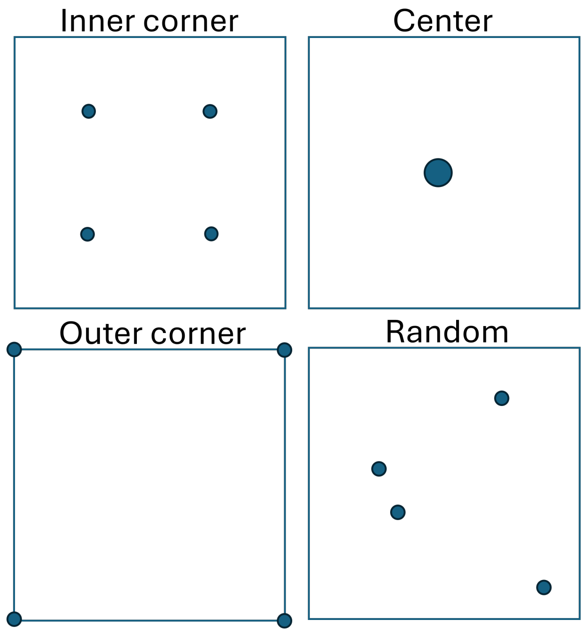

Figure 2.2: Peer instruction seating arrangements. Circles denote high aptitude students. The inner corner configuration places high aptitude students halfway between the center and the corner of the classroom. The outer corner configuration places high aptitude students at the corner of the classroom. The center configuration places high aptitude students in the center of the classroom. The random configuration places high aptitude students randomly throughout the classroom.

each trial was obtained by using a Levenberg-Marquardt algorithm to fit a power law ($y = ax^m$) to the fraction of learned students as a function of the generation number. We only considered the first 50% of the data for the PI model or the first 25% of the data for the traditional model. This truncation was done so that we only fit the part of the data before the finite size effect starts to affect the simulation.

2.3 The binary probabilistic cellular automata model for a traditional classroom set up

- Governing equation:

$$P_{ij} = \lambda_0 \tag{2.3}$$

where:

$P_{ij} \in [0, 1]$ is the probability of the student seated in row i and column j to learn,

$\lambda_0 \in [0, 1]$ is the probability of the student i, j to learn from the teacher

- Other relevant rules:

    1. All students are unlearned at the start of the simulation.
    2. Simulation is considered done when all students are learned.

2.4 Results: PI vs Traditional

- List of input and output parameters?
- m vs λ or ρ_0
- t_{max} vs λ or ρ_0
- t_{max} vs N for specific λ or ρ_0
- Comparison between levels of homogeneity of learning rates

2.5 Discussion/conclusions?

1. The traditional learning model is more scalable. Between the same ρ, the power law exponent b is lower than its peer-to-peer counterpart.
2. Inner corner configurations have higher b-values, while the traditional configurations have lower b-values.
3. B-values generally increases with ρ values
4. Intersections where PI is more efficient occur at lower class sizes and lower ρ values.
    - In some class sizes, traditional and P2P approaches can become equally efficient depending on the learning rate ρ.
5. Similar finding with previous research [2]: Students with less background knowledge learned as much with PI as students with more background knowledge with traditional instruction.

Chapter 3

Customized Chapter

This is a customized chapter.

3.1 First section

You may change the filename of this file as long as you correspondingly change the filename stated in the `input{}` line in the `main.tex`.

3.2 Including figures

Figures can be added into your LaTeXfiles using the `figure` environment. However, it is **recommended** that you use `.eps` file format. These are encapsulated postscript files. This can be easily done by installing a postscript printer that outputs to a file (port is FILE). Ask your system administrator to install such a device for producing `eps` files by simply printing it to that device.

Shown in Fig. 3.1 is a figure created from Excel®. Note that the bounding box and page bounding box should be adjusted well enough to show the correct field.

Please see the sample chapter on how the figure is included in the `tex` files.

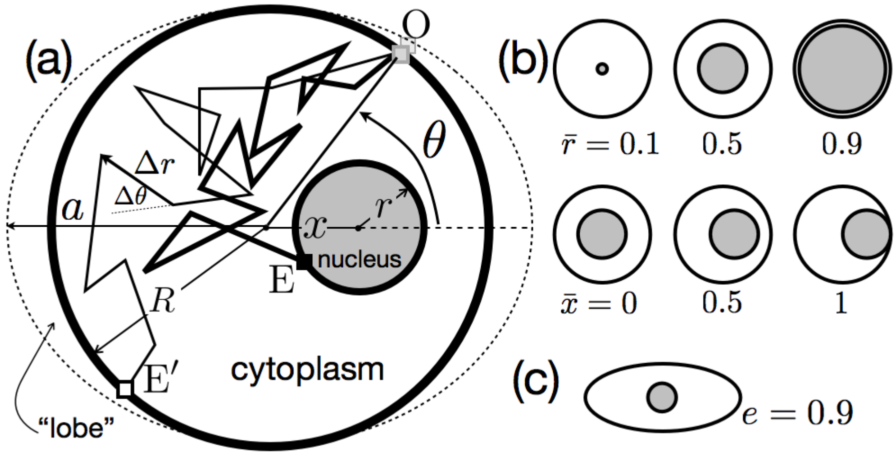

Figure 3.1: A *.png file.

Chapter 4

Another Customized Chapter

In this chapter, a sample for making equations and sub-equations are demonstrated.

4.1 Equations and sub-equations

In the following, a set of equations is shown.

$$\vec{\nabla} \cdot \vec{D} = \rho \tag{4.1a}$$

$$\vec{\nabla} \cdot \vec{B} = 0 \tag{4.1b}$$

$$\vec{\nabla} \times \vec{E} = -\partial_t \vec{B} \tag{4.1c}$$

The last one being

$$\vec{\nabla} \times \vec{H} = \vec{J} + \partial_t \vec{D} \tag{4.1d}$$

Note that text can still be placed between sub-equations within the `subequations` environment.

When using a solitary equations, you may use the usual equation syntax in LaTeX.

$$E = mc^2 \tag{4.2}$$

Chapter 5

Summary and Conclusions

A short sample thesis/dissertation is presented. Although not complete, it will be useful for newbies in LaTeX. Any questions? email me at the following address: `johnrob.bantang@gmail.com`.

Appendix A

Sample appendix

This gives an example of an appendix chapter. Note that this file has been included **after** the line `\appendix` in `main.tex`.

A.1 Equations in appendix

You don't need to worry about equations within the appendix since LaTeXautomatically formats the equation numbers for you. For example,

$$c^2 = a^2 + b^2 \tag{A.1}$$

becomes the Pythagorian theorem where c is the length of the longest side of any right triangle.

A.2 Codes as appendix

Include your codes when necessary to your thesis/dissertation. To do this, you may use `verbatim` environment as follows. **WARNING:** All verbatim and verbatiminput environments should always be treated as a separate paragraph. When included in a text paragraph, it sometimes happen to reduce the 1.5 spacing to the usual single-spaced text.

```
#include <iostream>
using std::cout;
using std::endl;

int main( void )
{
   cout << "Hello world!" << endl;
   return 0;
}
```

The {\small } bracketed region is used to lower the font size of the entire verbatim text. This will save you much space and give a more aesthetical look in your manuscript.

On the other hand, when very long codes are wished to be included automatically without the tedious cut and paste procedure, you may include them using the \verbatiminput command as follows. You may want to include a short description of the code of course.

```
//Johnrob Y. Bantang, Natinal Institute of Physics
//Created: 03 October 2002
// Makes new C files
// usage: newC filename
//Modifications:
// >> 21 Jan 2003, Johnrob
//   included the constant AUTHOR and AFFILIATION for portability

#include <stdlib.h>
#include <iostream.h>
#include <fstream.h>
#include <strstream.h>
#include <time.h>
#include <string.h>

const char *const AUTHOR= "Johnrob Y. Bantang";
const char *const AFFILIATION= "National Institute of Physics";
const char *const EXTENSION= ".cpp";

int main(int argc,char **argv){
if(argc!=2){
cout<<"usage: newC filename"<<endl;
exit(0);
}
char *fname= new char[strlen(argv[1])+5];
ostrstream out(fname,strlen(argv[1])+5);
out<<argv[1]<<EXTENSION;
time_t date=time(NULL);

ofstream file(fname,ios::nocreate,0);
//opens normal file that **already exists**;
if(file){
for(int i=1;i<5;i++)
cout<<"WARNING! file aready exists!"<<endl;
cout<<endl<<"you can type"<<endl<<endl;
cout<<"\t\"head "<<fname<<"\""<<endl;
cout<<"in command line to *view version*"<<endl<<endl;
cout<<"please enter 1 to OVERWRITE this file"<<endl;
```

```
cout<<"type anything to cancel"<<endl;
for(int i=1;i<5;i++)
cout<<"WARNING! file aready exists!"<<endl;
int n;
cin>>n;
if(n!=1){
cout<<"*no* file is created... exiting..."<<endl<<endl;
exit(0);
}
file.close();
file.open(fname);
if(!file)
cout<<"**cannot create new file!!**"<<endl;
cout<<"OLD FILE: "<<fname<<" *overwritten!*"<<endl;
}
if(!file){
file.open(fname);
if(!file){
cout<<"**cannot create new file!!**"<<endl;
exit(0);
}
cout<<endl<<"NEW FILE created: "<<fname<<endl<<endl;
}

cout<<"\tcreating contents for the new C++ file: "<<endl<<endl;
//creating headers...
file<<"//filename: \""<<fname<<"\""<<endl;
file<<"//"<<AUTHOR<<", "<<AFFILIATION<<endl;
file<<"//Created: "<<asctime( localtime(&date) );
//writes the time and date today; endl already in asctime();
file<<"//"<<endl;
file<<"//Comments:"<<endl;
file<<"// >>"<<endl;
file<<"//"<<endl;
file<<"//This file is generated using the \"newC generator\"..."<<endl;
file<<"//Modifications:"<<endl;
file<<"// >>"<<endl<<endl;
file<<"#include <iostream.h>"<<endl;
file<<"#include <math.h>"<<endl<<endl;
//starting the main body...
file<<"int main(int argc,char **argv){"<<endl;
file<<"//\tif(argc!= ...){"<<endl;
file<<"//\t\tcout<<\"usage: "<<argv[1]<<".exe ... \"<<endl;"<<endl;
file<<"//\t\texit(1);"<<endl;
file<<"//\t}"<<endl;
file<<"\t//write the main body here"<<endl;
file<<"return(0);"<<endl<<"}"<<endl;

delete fname;
```

```
  file.close();
  cout<<endl<<"\tCREATION SUCCESSFUL!"<<endl<<endl;
return 0;
}
```

This time, you may just include your recent codes by just copy-paste-ing the codes (as long they are clean!) into the directory `codes/` in the directory where this file is saved.

Bibliography

[1] M Arciaga, M Pastor, R Batac, J Bantang, and C Monterola. Experimental observation and an empirical model of enhanced heap stability resulting from the mixing of granular materials. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(07):P07040, 2009.

[2] Nathaniel Lasry, Eric Mazur, and Jessica Watkins. Peer instruction: From harvard to the two-year college. *American journal of Physics*, 76(11):1066–1069, 2008.

[3] Pierre-Yves Louis and Francesca R Nardi. *Probabilistic Cellular Automata: Theory, Applications and Future Perspectives*, volume 27. Springer, 2018.

[4] RM Roxas, SL Carreon-Monterola, and C Monterola. Seating arrangement, group composition and competition-driven interaction: Effects on students' performance in physics. In *AIP Conference Proceedings*, volume 1263, pages 155–158. American Institute of Physics, 2010.