Wildfire Simulation

Name: Steven Lo

Date: February 21st, 2025

CSCI-UA 144 MATH-UA 330: Introduction to Computer Simulation

MATLAB Code: https://github.com/stevloc/matlab/tree/main/Wildfire%20Simulation

Personal Note

The inspiration for this project stems from the devastating wildfires that have recently ravaged regions like Los Angeles. These real-life events highlight the destructive power of wildfires and the urgent need for better understanding and prediction of fire behavior. In Los Angeles, rapid urban expansion into wildland areas has increased the risk of catastrophic fires threatening biodiversity and global ecosystems. By simulating fire propagation using a cellular automaton model, this project aims to explore the fundamental dynamics of wildfire spread, providing insights that could inform fire management strategies, improve risk assessment, and ultimately help mitigate the impact of these disasters in vulnerable regions.

Introduction

Wildfires are complex natural events that spread rapidly, consuming vegetation and reshaping ecosystems worldwide. Predicting and understanding their behavior is essential for effective wildfire management and prevention. This project aims to simulate fire spread in a forest using a cellular automaton, a mathematical model that represents large-scale processes through simple, localized interactions (Von Neumann, 1966). By applying a set of predefined rules, the model determines how fire moves through a two-dimensional grid over discrete time steps.

A cellular automaton (CA) is a system in which a grid of cells updates its state based on a set of rules and the influence of neighboring cells. Cellular automata are commonly used in scientific simulations to model various natural and physical processes, including fluid flow, disease transmission, traffic systems, and chemical reactions (Grimmett, 1999). In this case, each cell in the grid represents a tree, and the fire spreads according to specific transition rules. At any given moment, each tree exists in one of three states: unburned, burning, or burned. A tree that is unburned can ignite if exposed to fire, a burning tree actively spreads fire to its neighbors, and a burned tree remains in its final state without further interaction. This structure allows the simulation to represent how wildfires progress over time and how different conditions influence their spread.

The model follows the Greenberg-Hastings cellular automaton, which is commonly used to simulate excitable media (Greenberg & Hastings, 1978). An excitable medium consists of elements that transition through different stages of activity, usually cycling from an inactive state to an excited state and finally to a refractory state before stabilizing. This framework is widely applied in fields such as neuroscience, cardiac wave modeling, and chemical pattern formation. In the context of fire spread, it ensures that trees burn for a limited time before becoming permanently burned, preventing unrealistic scenarios where trees reignite indefinitely.

Fire propagation in this model is determined by the Von Neumann neighborhood, which restricts interactions to the four adjacent cells: above, below, left, and right (Von Neumann, 1966). This contrasts with the Moore neighborhood, which includes diagonal neighbors as well. Using the Von Neumann neighborhood provides a more structured and predictable pattern of fire spread, similar to how real wildfires move along fuel-rich paths rather than instantly spreading in all directions. The limitation to four neighboring cells ensures that fire expansion remains controlled and dependent on direct adjacency rather than diagonal connectivity.

To introduce an element of reality, the model incorporates probabilistic fire spread, meaning that even when a tree is next to a burning tree, it does not necessarily catch fire (Stauffer & Aharony, 1994). Instead, fire spreads based on a predefined fire spread probability, which represents the likelihood that an unburned tree will ignite when exposed to flames. This probabilistic approach prevents fire from spreading uniformly and allows for more natural-looking fire progression patterns. Depending on the spread probability, the fire can either die out quickly, burn in patches, or consume the entire forest.

The objective of this simulation is to analyze how different fire spread probabilities affect wildfire propagation. By adjusting the probability of fire transmission, different outcomes can be observed. A low probability results in minimal fire expansion, with the fire struggling to sustain itself. A moderate probability allows fire to spread across much of the forest while still leaving some trees unburned. A high probability leads to rapid and widespread destruction, where nearly all trees are consumed by the fire. Through these variations, the simulation provides insight into how localized tree-to-tree interactions contribute to large-scale wildfire behavior. While the model does not account for environmental factors such as wind, humidity, or terrain, it serves as a foundational approach for understanding fire dynamics and lays the groundwork for more complex simulations.

Equations

This simulation models the spread of fire through a two-dimensional forest grid using a cellular automaton approach. Instead of relying on traditional differential equations, the fire spreads based on a set of discrete rules that determine how each tree changes state over time. These rules define the system's behavior and serve as our governing equations in an algorithmic form. At any given time, each tree (represented as a cell in the grid) exists in one of three states:

- Quiescent (0) – The tree is unburned and can catch fire.
- Excited (1) – The tree is currently burning.
- Refractory (2) – The tree has burned and will remain in this state permanently.

The fire spreads based on interactions between neighboring trees, which are defined using a Von Neumann neighborhood (only up, down, left, and right). This means that a burning tree can ignite only its four adjacent neighbors. The fire spread rule is given by Equation 1.

$$F(i,j)_{\text{new}} = 1, \quad \text{if at least one neighbor of } (i,j) \text{ is burning and } P_{\text{spread}} > \text{rand}()$$

Equation 1. Fire Spread Probability Rule

In this Equation, $F(i, j)$ represents the state of a tree at position (i, j) in the grid, and P_{spread} is a fixed probability that controls whether fire spreads. The function rand() generates a random number between 0 and 1, meaning that the fire does not always spread but does so in a way that introduces some randomness. Once a tree starts burning, it transitions to the refractory state in the next timestep. (see Equation 2)

$$F(i, j)_{\text{new}} = 2, \quad \text{if } F(i, j)_{\text{old}} = 1$$

Equation 2. Burning Tree Transition Rule

This ensures that a burning tree does not remain burning indefinitely—it burns for only one time step before becoming permanently burned. The simulation also tracks the total elapsed time, assuming that each timestep represents one hour of real-world time. (see Equation 3)

$$T_{\text{total}} = \text{clock} \times \Delta t$$

Equation 3. Total Time Elapsed

In Equation 3, clock is the number of simulation steps and Δt is set to 1 hour per step. Additionally, at each timestep, the simulation computes the number of trees in each state. (see Equation 4)

$$N_{\text{unburned}} = \sum F(i, j) == 0, \quad N_{\text{burning}} = \sum F(i, j) == 1, \quad N_{\text{burned}} = \sum F(i, j) == 2$$

Equation 4. Conservation of Total Trees

In Equation 4, N represents the count of trees in each category. These values are displayed in real time to visualize the fire's progress. The model starts with a single burning tree at the center of the grid. (see Equation 5)

$$F\left(\frac{n}{2}, \frac{m}{2}\right) = 1$$

Equation 5. Initial Fire Ignition Condition

This ensures that the fire spreads outward naturally, following the governing equations described above.While this model is simple, it captures key aspects of fire dynamics, including fire propagation, random variation, and irreversible burning. However, it does not account for factors like wind, terrain, or fuel density, which would require more complex differential equations. Despite these limitations, this discrete-state model effectively demonstrates how a fire spreads over time and eventually burns out.

Numerical Method

The fire spread simulation does not rely on traditional differential equations, so there is no need to approximate derivatives using numerical methods like finite differences. Instead, it is based on a discrete-time cellular automaton model, where time progresses in fixed steps and the system evolves according to a set of rules rather than continuous differential equations. Each time step, t, updates the state of each cell in the grid based on the state of its neighbors in the previous time step t. The core update rule follows Equation 6.

$$F(i,j)^{t+1} = \begin{cases} 1, & \text{if at least one neighbor of } (i,j) \text{ is burning at } t \text{ and } P_{\text{spread}} > \text{rand}() \\ 2, & \text{if } F(i,j)^t = 1 \text{ (burning tree turns into burned state)} \\ F(i,j)^t, & \text{otherwise} \end{cases}$$

Equation 6. Fire Spread Rule

In this equation, $F(i,j)^t$ represents the state of a cell at position (i,j) at time step t and $F(i,j)^t$ is its state in the next time step. The function rand() generates a random number between 0 and 1, ensuring that fire spreads probabilistically rather than deterministically. The total elapsed time in the simulation is given by Equation 3.

$$T_{\text{total}} = \text{clock} \times \Delta t$$

Equation 3. Total Time Elapsed

From Equation 3, each time step represents a fixed interval $\Delta t = 1$ hour. This means that the simulation progresses in discrete jumps of one hour per step rather than continuously. Since the governing equations are already discrete and require no numerical approximation, the numerical method directly follows the state transition rules implemented in the code. There is no need for numerical solvers or approximations, as all calculations involve simple conditional updates at each step.

Validation

To confirm that the simulation correctly follows the governing equations, we validate it using two controlled test cases: a deterministic fire spread test and a conservation of total trees check.

Deterministic Fire Spread Test

To ensure that fire spreads correctly according to the model's rules, the spread probability is set to P_{spread} = 1. This removes randomness, making fire spread deterministic—every unburned tree adjacent to a burning tree must catch fire in the next time step. For a small 7x7 grid, with the fire starting at the center, the expected fire pattern after three time steps is displayed by Equation 7.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 2 & 2 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Equation 7. Expected Fire Spread Pattern Validation in a 7x7 grid

Running the simulation under these conditions produces the same expected pattern, confirming that fire moves only to adjacent cells, does not skip any unburned trees, and correctly transitions from burning to burned.

Conservation of Total Trees Check

Since the simulation runs on a fixed-size grid, the total number of trees remains constant at all times. At any time step t, the sum of unburned, burning, and burned trees satisfies Equation 8.

$$N_{\text{unburned}}(t) + N_{\text{burning}}(t) + N_{\text{burned}}(t) = n \times m$$

Equation 8. Conservation of Trees in a Fixed Grid

The simulation tracks these counts at each time step, and the condition holds throughout, verifying that no trees are unintentionally created or removed from the system.

Interesting Aspects of the Code

The code effectively models fire spread using a cellular automaton approach, where each tree is represented as a cell in a grid and transitions between states based on predefined rules. The simulation operates with a simple three-state system: unburned (0), burning (1), and burned (2). This structure allows the model to capture essential fire dynamics while maintaining computational efficiency.

One of the key features of the code is its probabilistic fire spread mechanism. At each time step, the simulation evaluates whether a burning tree ignites its unburned neighbors based on a spread probability, P_{spread}. This is implemented using a random number generator, ensuring that fire spread is not entirely deterministic, adding variation and making each simulation run unique. The core fire spread logic is implemented as it is seen by Figure 1.

```
for t = 1:size(neighbor_offsets, 1)
    ni = i + neighbor_offsets(t, 1);
    nj = j + neighbor_offsets(t, 2);

    if ni >= 1 && ni <= n && nj >= 1 && nj <= m
        if Forest(ni, nj) == 0 && rand() < spread_prob
            Forest_copy(ni, nj) = 1;
        end
    end
end
```

Figure 1. Core for Fire Spread Logic Implementation

Another important aspect in Figure 1 is the use of a Von Neumann neighborhood model, which means fire spreads only to the four directly adjacent cells (up, down, left, and right). This restriction prevents unrealistic diagonal fire expansion and results in a more natural outward spread. Since all cells update simultaneously at each time step, the simulation ensures synchronized fire progression across the grid. The code also includes an automatic termination condition, stopping the simulation once all burning trees have either spread the fire or burned out. From Figure 2, this prevents unnecessary computations and ensures efficiency.

```
if num_burning == 0
    break;
end
```

Figure 2. Code for Fire Termination Condition

To enhance visualization and track fire behavior, the code continuously updates the counts of unburned, burning, and burned trees at each step. These values are displayed in real-time, providing a clear representation of how the fire evolves, which can be demonstrated by Figure 3.

```
num_unburned = sum(Forest(:) == 0);
num_burning = sum(Forest(:) == 1);
num_burned = sum(Forest(:) == 2);

imagesc(Forest);
title({['Time Step: ', num2str(clock), ' hours elapsed (T_{total} = ', num2str(T_total), ' hours)']; ...
       ['Unburned: ', num2str(num_unburned), ...
       ' | Burning: ', num2str(num_burning), ...
       ' | Burned: ', num2str(num_burned)]});
```

Figure 3. Code for Fire State Visualization and Tracking

The total elapsed time is tracked using a simple time-step calculation is displayed below in Figure 4.

```
T_total = clock * delta_t;
```

Figure 4. Code for Time-Step Calculation for Simulation Progression

The code from Figure 4 ensures that the fire's duration is correctly represented in hours. These elements combine to create a computationally efficient and effective fire spread model that balances simplicity with meaningful simulation results.

Results and Discussion

For this simulation, a 100x100 grid will be used, meaning that in total there will be 10,000 trees alive at the beginning of the simulation. Additionally, the total time step will be 100 (Total Simulation Time is 100 hours). Its results highlight how fire propagates through the forest under different fire spread probabilities P_{spread}. The key metrics analyzed were the number of unburned, burning, and burned trees over time. The three cases below illustrate distinct fire spread behaviors at 100 hours into the simulation, demonstrating the effect of different values of P_{spread}: 0.5, 0.75, and 1.0.

Limited Fire Spread ($P_{spread} = 0.5$)

In the first case, with $P_{spread} = 0.5$, the fire spreads slowly and burns only 775 trees, with 14 still burning at the 100-hour mark. The majority of the forest (9,211 trees) remains unburned. The fire forms a patchy and irregular shape, indicating that fire propagation was frequently interrupted. This behavior is expected in real wildfire scenarios where low fire intensity, high humidity, or natural firebreaks prevent continuous spread. The low probability of ignition leads to isolated burn areas rather than a fully connected fire front. (see Figure 5)

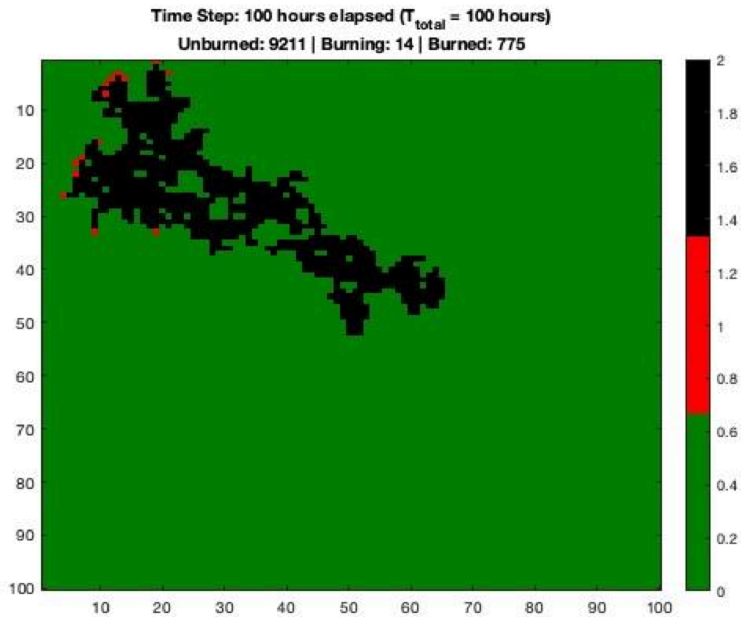

Figure 5. Limited Fire Spread with P_{spread} = 0.5

Extensive Fire Spread with Scattered Unburned Trees (P_{spread} = 0.75)

In the second case, with P_{spread} = 0.75, fire spreads more aggressively, consuming 9,937 trees, with only 60 trees remaining unburned. A few trees survived, scattered across the grid, suggesting that randomized burning patterns left small patches untouched. This behavior is typical of moderate-intensity wildfires, where fire spreads efficiently but some areas escape burning due to variations in fuel availability or environmental conditions. (see Figure 6)

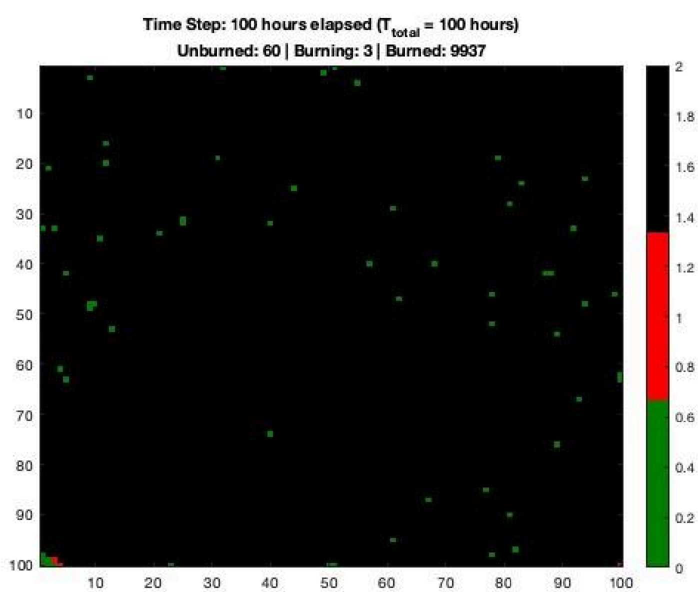

Figure 6. Extensive Fire Spread with P_{spread} = 0.75

Complete Forest Burn (P_{spread} = 1.0)

The third case represents an extreme scenario where P_{spread} = 1.0, meaning every unburned tree adjacent to fire catches fire with certainty. At 100 hours, 9,999 trees have burned, with only 1 tree still burning. There are no unburned trees left, indicating unrestricted fire propagation. This outcome occurs in severe wildfire conditions where drought, high winds, and abundant dry vegetation allow flames to move rapidly without interruption, leading to near-total destruction. (see Figure 7)

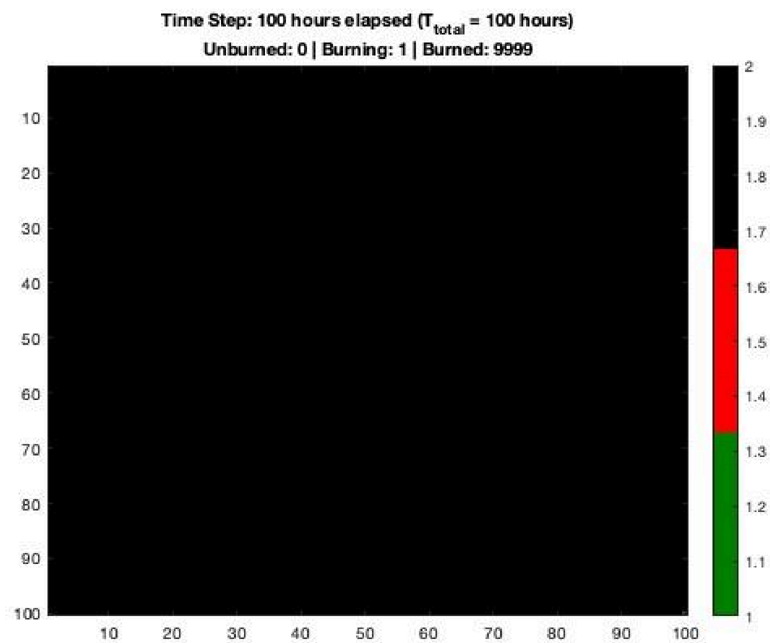

Figure 7. Complete Forest Burn with P_{spread} = 1.0

These results emphasize how fire spread probability significantly affects the extent of destruction. At lower probabilities, fire struggles to sustain itself, resulting in small burn areas and many untouched trees. With higher probabilities, fire spreads efficiently, burning most or all of the forest. The model effectively captures the stochastic nature of wildfire propagation, demonstrating how small changes in fire behavior can lead to vastly different outcomes. To better illustrate the impact of fire spread probability on wildfire propagation, Figure 8 presents a visual representation of fire progression on a 50x50 forest grid under three different spread probabilities: P_{spread} = 0.5, 0.75, and 1.0. Each cell in the grid represents a tree, with green indicating unburned trees, red representing actively burning trees, and black showing trees that have already burned. (see Figure 8)

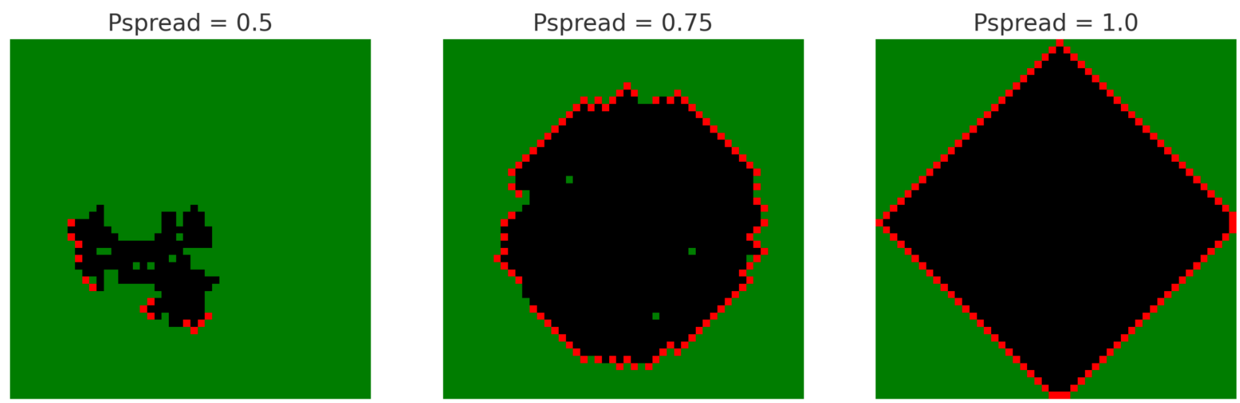

Figure 8. Fire Spread Simulation at Different Spread Probabilities

The results demonstrate how variations in P_{spread} affect the extent and pattern of fire expansion. At P_{spread} = 0.5, fire propagation is limited and irregular, forming isolated burn patches while leaving a significant portion of the forest untouched. This reflects real-world scenarios where fire intensity is low or environmental conditions, such as humidity or natural barriers, prevent continuous spread. When P_{spread} increases to 0.75, fire spreads more extensively, consuming most trees while leaving a few unburned areas scattered across the grid. This pattern represents moderate-intensity wildfires where fire is still somewhat unpredictable, but large portions of the forest succumb to burning. At the extreme end, P_{spread} = 1.0 leads to unrestricted fire expansion, where nearly all trees are consumed, leaving behind only burned remains. This scenario closely resembles large-scale wildfires under dry, wind-driven, and fuel-rich conditions, where flames spread rapidly without interruption.

These simulations highlight how small changes in fire spread probability can significantly alter wildfire outcomes, demonstrating the nonlinear nature of fire dynamics. In real-world wildfire management, understanding these probabilistic behaviors is crucial for predicting fire risks and implementing effective control strategies. The results emphasize how minor shifts in environmental conditions or ignition probability can determine whether a fire remains a contained burn or escalates into a full-scale disaster.

Conclusions

This project explored the spread of fire in a two-dimensional forest grid using a cellular automaton model. The simulation applied a three-state system where trees transitioned from unburned to burning and finally to a permanently burned state. Fire propagation was governed by a probabilistic rule that determined whether a burning tree would ignite its neighbors, with different spread probabilities significantly affecting the extent and speed of the fire.

Through multiple simulations, we observed that low spread probabilities (P_{spread} = 0.5) resulted in minimal fire expansion, often leaving large sections of the forest untouched. As the probability increased (P_{spread} = 0.75), the fire spread more aggressively, burning most of the forest while leaving a few scattered unburned trees. When the spread probability reached 1.0, the fire consumed nearly the entire forest, demonstrating the impact of an unrestricted wildfire scenario.

This project reinforced how simple mathematical rules can model complex natural phenomena. It also demonstrated how small changes in parameters can lead to vastly different outcomes, mirroring real-world wildfire behavior. Additionally, the use of a discrete time-step model allowed for effective tracking of fire progression over time. The simulation successfully provided insight into fire dynamics, showing how local interactions can lead to large-scale patterns in fire spread.

References

Bak, P., Tang, C., & Wiesenfeld, K. (1987). *Self-organized criticality: An explanation of 1/f noise. Physical Review Letters, 59*(4), 381-384. https://doi.org/10.1103/PhysRevLett.59.381

Greenberg, J. M., & Hastings, S. P. (1978). *Spatial patterns for discrete models of diffusion in excitable media. SIAM Journal on Applied Mathematics, 34*(3), 515-523. https://doi.org/10.1137/0134040

Grimmett, G. (1999). *Percolation.* Springer-Verlag.

Stauffer, D., & Aharony, A. (1994). *Introduction to percolation theory.* CRC Press.

Von Neumann, J. (1966). *Theory of self-reproducing automata* (A. W. Burks, Ed.). University of Illinois Press.