Modelling and Simulation of Laser Dynamics with Cellular Automata on Parallel and Distributed Computers

José Luis Guisado Lizar

Advisors:
- Francisco Jiménez Morales (Universidad de Sevilla)
- Francisco Fernández de Vega (Universidad de Extremadura)

Committee:
- Antonio Córdoba Zurita (Univ. Sevilla)
- Marco Tomassini (Univ. Lausanne)
- Alfons Hoekstra (Univ. Amsterdam)
- Juan Julián Merelo Guervós (Univ. Granada)
- Fernando Díaz del Río (Univ. Sevilla)

Departamento de Física de la Materia Condensada
Universidad de Sevilla

14 December 2009

QUESTION 1

LASER

Mechanism
- Stimulated emission
  \[ E = h\nu \]

Phenomenology
- Laser threshold
- Relaxation oscillations

Is it possible to model a laser using a cellular automaton (CA) that reproduces the phenomenology?
Interest of a CA laser model

- New methodological approach
- Reinforces the vision of laser as a complex system

Applications:
- Lasers ruled by equations with convergence problems
- Difficult boundary conditions
- Very small optoelectronic devices
- Study interesting problems: chaos or emergent phenomena in lasers
- Parallel model

QUESTION 2

Is it possible to take advantage of the intrinsic parallel nature of CA to develop efficient parallel implementations?
Modelling laser dynamics with a cellular automaton
1. Introduction: Cellular Automata (CA) and laser dynamics
2. CA model for laser dynamics
3. Simulation results

Efficient implementation on parallel and distributed computers
1. Parallelization
2. Execution on dedicated clusters
3. Execution on non-dedicated clusters with load balancing

Conclusions

Cellular automata (CA)

A class of mathematical systems:
- Space and time: discrete
- Each cell: discrete states
- Local interactions: each cell interacts only with its neighbours
- Discrete dynamics: evolution rules
- Parallel nature

Models for complex systems $\rightarrow$ emergent behaviours

Applications:
- Natural sciences: models in physics, chemistry, biology, geology...
- Mathematics
- Theoretical computer science
- Engineering
Laser: physical processes

- **Laser**: Device that generates electromagnetic radiation based on the stimulated emission process:

![Laser Diagram]

- This process competes with absorption
- Normally: lower level more populated \(\Rightarrow\) absorption has greater probability than emission
- Laser mechanism: energy pumping process \(\Rightarrow\) population inversion
- An incoming photon with \(h\nu=E_{12}\) can give rise to a cascade of stimulated coherent photons

Standard description of laser dynamics: laser rate equations

\[
\begin{align*}
\frac{dn(t)}{dt} &= K N(t) n(t) - \frac{n(t)}{\tau_c} \\
\frac{dN(t)}{dt} &= R - N(t) \frac{N(t)}{\tau_a} - K N(t) n(t)
\end{align*}
\]

- \(n(t)\) → number of laser photons
- \(N(t)\) → population inversion
- \(\tau_c\) → decay time of photons in the cavity
- \(\tau_a\) → decay time of the upper laser level
- \(R\) → Pumping rate
- \(K\) → Coupling constant
Laser dynamics: Dependence of threshold pumping rate on laser parameters

- From the laser rate equations:
  \[ R_i = \frac{1}{K \tau_a \tau_c} \]

Laser dynamics: behaviours

- **Laser operation** → different characteristic **dynamic behaviours** are shown depending on the type of laser (or on the laser parameters):
  - **Constant regime:**
    - After an initial transient, the system reaches a stationary constant behaviour
  - **Relaxation oscillations (or laser spiking):**
    - Correlated large amplitude damped oscillations in the photon number and population inversion
Laser dynamics: Dependence of behaviour on laser parameters

- **Laser rate equations** → depending on parameters values, 2 main behaviours:
  - Oscillatory
  - Constant regime

Theoretical stability curve:

\[
\frac{\tau_s}{\tau_c} = \frac{\left( \frac{R}{R_s} \right)^2}{4 \left( \frac{R}{R_s} - 1 \right)}
\]

- \( \tau_s \) → Life time of excited electrons
- \( \tau_c \) → Life time of laser photons
- \( R \) → Pumping rate

CA model for laser dynamics (1)

2D, multivariable and partially probabilistic CA:

- **Cellular space**: 2-dims. square lattice with periodic boundary conditions

- **States of the cells**: each cell has four variables associated:
  \[\begin{align*}
  a_r(t) &\in \{0,1\} \quad \to \text{State of the electron} \\
  c_j(t) &\in \{0,1,2,\ldots,M\} \quad \to \text{Number of photons} \\
  \bar{a}_r(t) &\in \{0,1,2,\ldots,\tau_a\} \quad \to \text{Time since electron in upper laser state} \\
  \bar{c}^k_r(t) &\in \{0,1,2,\ldots,\tau_c\} \quad \to \text{Time since photon } k \text{ was created}
  \end{align*}\]

  \( (\text{in cell } \vec{r} = (i, j) \text{ at time } t) \)

- **Neighbourhood**: "Moore neighbourhood": Each cell has nine neighbours:

\[
\Gamma_r(t) = \sum_{j'=\text{neighbo}(r)} c_{j'}(t)
\]
CA model for laser dynamics (2)

Transition function:

- **R1- Pumping:** If \( a_r(t) = 0 \) \( \rightarrow a_r(t+1) = 1 \) with a probability \( \lambda \)
- **R2- Stimulated emission:** If \( \{ a_r(t) = 1, \Gamma_r > \delta \} \) \( \rightarrow \begin{cases} c_r(t+1) = c_r(t)+1 \\ a_r(t+1) = 0 \end{cases} \)
- **R3- Photon decay:** Photon is destroyed \( \tau_c \) time steps after it was created
- **R4- Electron decay:** Electron decays \( \tau_a \) time steps after it was promoted
- **R5- Evolution of temporal variable \( \tilde{a}_r(t) \):** counts number of time steps since an electron is promoted to upper state.
- **R6- Evolution of temporal variable \( \tilde{c}_r(t) \):** counts number of time steps since a photon is created.
- **R7- Random noise photons:** \( c_j(t+1) = c_j(t)+1 \) for \(-0.01\%\) of total cells

---

Simulations

- **Initial state:** \( a_r(0) = 0, c_j(0) = 0, \forall r \) except small fraction of noise photons
- The system evolves by the application of the transition rules
- In each time step, we measure:
  - \( n(t) \): Total number of laser photons
  - \( N(t) \): Total number of electrons in upper laser state \( \equiv \) population inversion
- System \( \rightarrow 3 \) parameters: \( \{ \lambda, \tau_c, \tau_a \} \):
  - \( \lambda \rightarrow \) Pumping probability
  - \( \tau_c \rightarrow \) Life time of laser photons
  - \( \tau_a \rightarrow \) Life time of excited electrons
- System size used: normally 400×400 cells
Simulation results:
Dependence of threshold pumping probability on laser parameters

- From the laser rate equations:
  \[ \lambda_0 = \frac{1}{C \tau_a \tau_c} \]  
  \((\lambda \propto R)\)

- From the simulation:

Simulation results:
Lasers behaviours

(a): Constant regime
(b): Relaxation oscillations (laser spiking)
Simulation results:
Dependence of behaviour on laser parameters

- **Laser rate equations** → depending on parameters values, 2 main behaviours:
  - Oscillatory
  - Constant regime
  \[ \tau_a = \frac{R}{R_0} \] (Theoretical stability curve)

- **Simulations** → Shannon's entropy of temporal distribution of n(t) and N(t): fingerprint of oscillations

\[ S = -\sum f_i \log_2 f_i \]

- \( \tau_a \) → Life time of excited electrons
- \( \tau_c \) → Life time of laser photons
- \( R \) → Pumping rate
- \( \lambda \) → Pumping probability
Simulation results:
Irregular oscillations

(c): Complex behaviour showing irregular oscillations:

Power spectrum: 1/f² noise
Simulations results: Spatio-temporal patterns

Oscillatory behaviour

Constant regime

Simulation of pulsed pumped lasers

- **Pumping rate** $R(t)$ → time dependent pulsed form:

  \[ R(t) = R_m \Phi(t) \]

  (Width: 2 $t_p$)

- **Laser rate equations:**

  \[
  \begin{align*}
  \frac{dn(t)}{dt} &= KN(t)n(t) - \frac{n(t)}{\tau_s} - \frac{\varepsilon N(t)}{\tau_s} \\
  \frac{dN(t)}{dt} &= R(t) - KN(t)n(t) - \frac{n(t)}{\tau_s}
  \end{align*}
  \]

  (\varepsilon → fraction of radiative decay processes that create a photon in the laser mode)
CA model for pulsed pumped lasers

- **Modified transition function:**
  
  **R1- Pumping:** If \( \{ a_r(t) = 0 \} \) \( \rightarrow \) \( a_r(t+1) = 1 \) with a probability: \( \lambda(t) = \lambda_{\text{max}} \Phi(t) \)
  
  **R2- Stimulated emission:** If \( \{ a_r(t) = 1, \Gamma_v > \delta \} \) \( \rightarrow \)
  \[
  \begin{cases}
  c_r(t+1) = c_r(t)+1 \\
  a_r(t+1) = 0
  \end{cases}
  \]
  
  **R3- Photon decay:** Photon is destroyed \( \tau_c \) time steps after it was created
  
  **R4- Electron decay:** Electron decays \( \tau_a \) time steps after it was promoted and a new photon will be created in that cell with a probability \( \theta \)
  
  **R5- Evolution of temporal variable**: \( \tilde{a}_i(t) \): counts number of time steps since an electron is promoted to upper state.
  
  **R6- Evolution of temporal variable**: \( \tilde{c}_j(t) \): counts number of time steps since a photon is created.
  
  No random noise photons are created

Spontaneous emission process is associated with the decaying of the population inversion.

Pulsed pumped lasers results: differential equations versus CA

**Numerical integration of differential equations:**

**CA model simulation:**

(Pumping: Value of R(t))

(Pumping: Number of electron cells pumped by rule R1 in each time step)
Outline

- Modelling laser dynamics with a cellular automaton
  1. Introduction: Cellular Automata (CA) and laser dynamics
  2. CA model for laser dynamics
  3. Simulation results

- Efficient implementation on parallel and distributed computers
  1. Parallelization
  2. Execution on dedicated clusters
  3. Execution on non-dedicated clusters with load balancing

- Conclusions

QUESTION 2

Is it possible to take advantage of the intrinsic parallel nature of CA to develop efficient parallel implementations?
CA models on parallel computers

CA models:

- Very suitable to be implemented efficiently on **parallel computers**:
  - **Intrinsic parallel nature**: Evolution rules $\rightarrow$ in parallel to all the cells
  - **Local nature**: Evolution rules $\rightarrow$ local

$\Rightarrow$ The system can be **split into partitions**:

- Run on different processors
- Communication flow between processors can be kept low

Parallel version of the CA laser model

- For **detailed** laser dynamics **simulations** (fine grained 2D or 3D models):
  Large execution time and memory required $\Rightarrow$ **parallel implementation is necessary**

- **Difficulties** for good performance: CA model **only partially uncoupled**:
  - Synchronous CA $\Rightarrow$ Nodes must exchange information after each time step
  - All the nodes must have finished an iteration before the next one can be started
  - Parallel application performance: limited by the slowest task

- Such a model $\rightarrow$ good performance on shared memory parallel computers

- **Our contribution**:

  ... Is it feasible to run a parallel version of the model on parallel and distributed computers? (On dedicated or heterogeneous non-dedicated clusters)
Parallel implementation

- For **distributed-memory parallel computers**, using **message passing** (PVM)
- Following:
  - Master-slave programming model
  - Data decomposition methodology for workload allocation

---

**1D Domain decomposition** (stripes):
- minimize number of send/receive calls

- Ghost cells.

- Laser CA model: **only neighbours' state needed**: \( c_x(t) \) (# of photons)
  - minimize communications
Execution of parallel model

- **Heterogeneous PC cluster** - 10 nodes Intel Pentium-4:
  - 6 "fast" nodes (2.7 GHz): simulations with 1 to 6 nodes
  - 4 "slow" nodes (1.8 GHz): simulations with more than 6 nodes
  - O.S: Linux (Rocks distribution, based on Red Hat)
  - RAM Memory: 512 MB each node
  - Communications: Fast-ethernet 100 Mbps switch

- **To evaluate performance:**
  - Running the same experiment for different:
    - Number of nodes
    - System sizes

Performance analysis: runtime

- **Runtime** of the experiments for 3 different system sizes:
Dedicated cluster: speedup

- **Speedup** with respect to the sequential program for 3 different system sizes:

\[
\text{speedup} = \frac{\text{runtime of the sequential version}}{\text{runtime of the parallel version}}
\]

![Graph showing speedup vs. number of processors for different system sizes.](image)

Analysis of execution

- High **computation-to-communication ratio** (~10 for slaves):

![Graph showing computation-to-communication ratio.](image)
An application is said to be scalable if:
- When the number of processors and the problem size are increased by the same factor, the running time remains the same

Results:
- Small overhead: 2% to 5% ⇒ good scalability for small clusters

Non-dedicated cluster

Two main differences respect to most previous implementations of CA-based models:

1. Modular approach: the model is executed on top of a dynamic load balancing tool
   - More flexibility than directly implementing the load balancing on the application

2. It is possible to migrate load to cluster nodes initially not belonging to the pool
Non-dedicated cluster

Dynamic load balancing

- Dynamic load balancing tool:
  - **DYNAMITE**:
    - Developed by the group: “Section Computational Science”, from the University of Amsterdam
    - Based on “Dynamic PVM”, a re-implementation of PVM that adds dynamic load balancing
    - It monitors the use of the cluster nodes (CPU, memory)
    - It dynamically migrates tasks when one of them gets over- or under-used, as defined by configurable threshold values
  - [http://www.science.uva.nl/research/scs/Software/dynamite](http://www.science.uva.nl/research/scs/Software/dynamite)
Non-dedicated cluster: Performance

- Experiments:
  - Parallel application: 6 computing nodes + 1 master node
  - Artificial load: from 0 to 5 nodes
  - 10 nodes available on the cluster

- Results:

<table>
<thead>
<tr>
<th>CONFIGURATION</th>
<th>EXECUTION TIME (s)</th>
<th>IMPROVEMENT (RATIO)</th>
<th>IMPROVEMENT (PERCENT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No load balancing with artificial load on 1-5 nodes</td>
<td>1895.08</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Load balancing with load on 1 node</td>
<td>384.59</td>
<td>4.93</td>
<td>80 %</td>
</tr>
<tr>
<td>Load balancing with load on 2 nodes</td>
<td>564.76</td>
<td>3.36</td>
<td>70 %</td>
</tr>
<tr>
<td>Load balancing with load on 3 nodes</td>
<td>611.12</td>
<td>3.10</td>
<td>68 %</td>
</tr>
<tr>
<td>Load balancing with load on 4 nodes</td>
<td>1595.75</td>
<td>1.19</td>
<td>16 %</td>
</tr>
<tr>
<td>Load balancing with load on 5 nodes</td>
<td>1833.82</td>
<td>1.03</td>
<td>3 %</td>
</tr>
<tr>
<td>No load, with or without load balancing</td>
<td>233.43</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>

Experiments:
- Parallel application: 6 computing nodes + 1 master node
- Artificial load: from 0 to 5 nodes
- 10 nodes available on the cluster

Execution progress for different load levels
Outline

- Modelling laser dynamics with a cellular automaton
  1. Introduction: Cellular Automata (CA) and laser dynamics
  2. CA model for laser dynamics
  3. Simulation results

- Efficient implementation on parallel and distributed computers
  1. Parallelization
  2. Execution on dedicated clusters
  3. Execution on non-dedicated clusters with load balancing

- Conclusions

Conclusions (1)

- CA model for laser dynamics simulation → alternative to differential equations

  1. It reproduces the phenomenology of laser dynamics
     - Threshold pumping, laser behaviours (constant regime, laser spiking, irregular oscillations), dependence on parameters

  2. Flexible and robust model:
     - Modified model reproduces the behaviour of pulsed pumped lasers

  3. New methodological approach:
     - Laser as a complex system
     - Applications: when rate equations → difficult to integrate or not applicable
     - Parallel model
Conclusions (2)

4. **Parallel implementation** using the message passing paradigm

5. **Dedicated clusters:**
   - Implementation takes a **good advantage of parallelization:**
     - In spite of frequent communications (CA model only partially uncoupled)
     - High computation-to-communication ratio
   - **Good performance**
   - **Good scalability** on small clusters

6. **Heterogeneous non-dedicated clusters:**
   - **Dynamic load balancing**
   - Artificial load
   - Performance improvement if free nodes
   - **It is feasible to run a parallel version of the model**

Future work

- **Simulations of specific laser systems:**
  - 3D model
  - Boundary conditions

- **Scalability** of the model for massive parallelism

- **Implementation using Grid Computing:**
  - **Desktop grid** implementation already done → Using BOINC system
  - Application selected by the **EDGES Project:**
    - "Enabling Desktop Grids for e-Science", 7th Framework Program project, European Union
Publications

JOURNAL ARTICLES


CONFERENCE PROCEEDINGS AND BOOK CHAPTERS (I)


Publications

CONFERENCE PROCEEDINGS AND BOOK CHAPTERS (II)

   Laser dynamics modelling and simulation: An application of dynamic load balancing of parallel cellular automata.
   In E. Cantú-Paz and F. Fernández de Vega, editors, *Parallel and Distributed Computational Intelligence*, Studies in Computational Intelligence. Springer Verlag, 2009. Accepted.

   Simulación de la dinámica del láser mediante un modelo de autómata celular.

   Simulación en paralelo de la dinámica láser.

   Rendimiento y escalabilidad de simulaciones en paralelo de la dinámica láser sobre un cluster incorporando balanceo dinámico de carga.

Breve resumen en español

Modelado y Simulación de la Dinámica del Láser con Autómatas Celulares en Computadores Paralelos y Distribuidos

José Luis Guisado Lizar

Directores:
- Francisco Jiménez Morales (Universidad de Sevilla)
- Francisco Fernández de Vega (Universidad de Extremadura)

Tribunal:
- Antonio Córdoba Zurita (Univ. Sevilla)
- Marco Tomassini (Univ. Lausanne)
- Allons Hoekstra (Univ. Amsterdam)
- Juan Julián Merelo Guervós (Univ. Granada)
- Fernando Díaz del Río (Univ. Sevilla)

Departamento de Física de la Materia Condensada
Universidad de Sevilla
Objetivos

1. Modelar un láser utilizando un autómata celular que reproduzca la fenomenología de su dinámica.

2. Aprovechar la naturaleza intrínsecamente paralela de los autómatas celulares para desarrollar una implementación paralela eficiente de dicho modelo, sobre computadores paralelos y distribuidos:
   - Cluster dedicado
   - Cluster heterogéneo no dedicado

Conclusiones (1)

- **Modelo de autómata celular (AC) para simular la dinámica láser → alternativa a las ecuaciones diferenciales**

  1. **Reproduce la fenomenología** de la dinámica láser:
     - Bombeo umbral, comportamientos del láser (régimen constante, oscilaciones de relajación o "laser spiking", oscilaciones irregulares), dependencia respecto a los parámetros del sistema

  2. **Modelo flexible y robusto**:
     - Un modelo modificado reproduce el comportamiento de los láseres de bombeo pulsado

  3. **Nuevo enfoque metodológico**:
     - **Láser como un sistema complejo**
     - **Aplicaciones**: cuando las ecuaciones de balance del láser → difíciles de integrar o no aplicables
     - **Modelo intrínsecamente paralelo**
Conclusiones (2)

4. **Implementación paralela** usando el paradigma de paso de mensajes

5. **Clusters dedicados:**
   - La implementación aprovecha la paralelización:
     - A pesar de las comunicaciones frecuentes (el modelo de AC es sólo parcialmente desacoplado)
     - Alta tasa computación-comunicación
   - **Buen rendimiento.**
   - **Buena escalabilidad** sobre clusters pequeños.

6. **Clusters heterogéneos no dedicados:**
   - **Balanceo dinámico de la carga**
   - Carga artificial
   - **Es factible ejecutar una versión paralela del modelo**