JÜLICH FORSCHUNGSZENTRUM

Pedestrian dynamics

Implementation and analysis of ODE solvers

September 30, 2010 | Timo Hülsmann

Pedestrian dynamics: Implementation and analysis of ODE solvers

Parts of this talk

- Part 1: Introduction to pedestrian dynamics
- Part 2: Preserving data locality with Space-Filling Curves (Excursus)
- Part 3: Implementation and analysis of ODE solvers

Pedestrian dynamics

Part I: Introduction to pedestrian dynamics

September 30, 2010 | Timo Hülsmann

Pedestrian dynamics

Introduction

- Why pedestrian dynamics?
- Classification of Models
- Generalized Centrifugal Force Model (GCFM)

Why pedestrian dynamics?

- Enhancement of safety in complex buildings and mass events
- Simulation and optimization of evacuations
- Improvement of comfort in public buildings (airports, railway stations, shopping malls, etc.)
- Minimal travel times and maximum capacities

Classification of Models

Macroscopic Models

- System is described by mean values of characteristics: Conservation laws for density, flow, etc.

Microscopic Models

- Each pedestrian is treated seperately.
- Can be space-continuous (system of ODEs) or space-discrete (cellular automaton).
- Can be rule-based or force-based.

Generalized Centrifugal Force Model (GCFM)

Forces

- Driving force (carries pedestrians to desired direction with desired speed)
- Repulsive forces from other predestrians and walls (to avoid collisions)
- All Forces add up to the complete force acting on the pedestrian:

$$\vec{F}_i = \sum_{i \neq j}^{N} \vec{F}_{ij}^{rep} + \sum_{B} \vec{F}_{iB}^{rep} + \vec{F}_i^{driv}$$

Forces

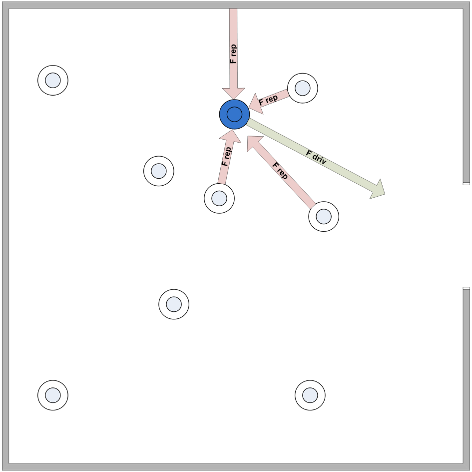


Figure: Example of forces acting on a pedestrian

JÜLICH FORSCHUNGSZENTRUM

Forces

- The repulsive force:

$$\vec{F}_{ij}^{rep} = -m_i K_{ij} \frac{(\eta V_i^0 + V_{ij})^2}{\text{dist}_{ij}} \vec{e}_{ij}\,, \quad \vec{R}_{ij} = \vec{R}_j - \vec{R}_i\,, \quad \vec{e}_{ij} = \frac{\vec{R}_{ij}}{\|\vec{R}_{ij}\|}$$

$$V_{ij} = \frac{1}{2}((\vec{V}_i - \vec{V}_j)\cdot\vec{e}_{ij} + |(\vec{V}_i - \vec{V}_j)\cdot\vec{e}_{ij}|)\,, \quad K_{ij} = \frac{1}{2}\frac{\vec{V}_i \cdot \vec{e}_{ij} + |\vec{V}_i \cdot \vec{e}_{ij}|}{\|\vec{V}_i\|}$$

- The driving force:

$$\vec{F}_{iB}^{driv} = m_i \frac{\vec{V}_i^0 - \vec{V}_i}{\tau}$$

JÜLICH FORSCHUNGSZENTRUM

Pedestrian dynamics

Part II: Preserving data locality with Space-Filling Curves (Excursus)

September 30, 2010 | Timo Hülsmann

Pedestrian dynamics

Data locality

- Idea: Reduce cache misses using data locality
- Space-Filling Curves
- Test cases
- Results
- Conclusion

Idea: Reduce cache misses using data locality

- We want to optimize the runtime. The aim is to simulate faster than realtime.
- In the implementation of the GCFM Linked-Cells are used to determine pedestrians' neighbourhood.
- Only the repulsive forces from pedestrians situated in the immediate neighbourhood are considered in the calculations.
- Data of nearby pedestrians should also be nearby in memory.
- Space-Filling Curves order 2-dimensional data s.t. locality is preserved.

Space-Filling Curves

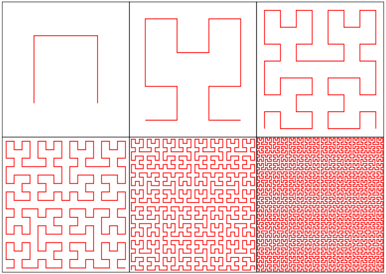
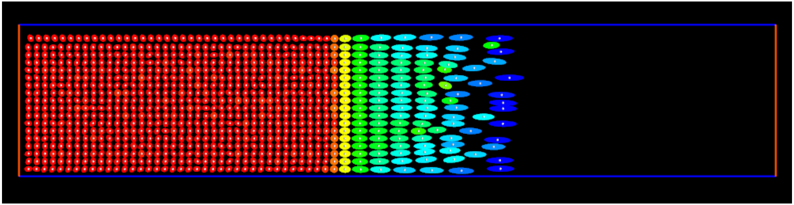


Figure:
http://upload.wikimedia.org/
wikipedia/commons/3/3a/Hilbert_curve.png

Hilbert curve

- Construction is a recursive process
- Continuous and surjective
- Good locality-preserving behavior

Test cases



- A corridor was used for testing and 1152 pedestrians were ordered
  - columnwise
  - rowwise
  - along a hilbert curve
  - random

Results

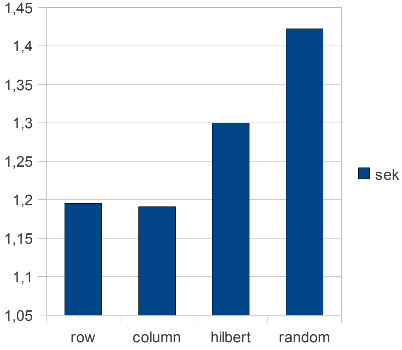


Figure: First time-step

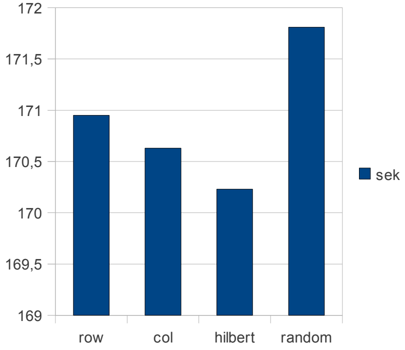


Figure: Complete simulation

Conclusion

- In the first time-step columnwise and rowwise ordering are faster than ordering along a hilbert curve.
- Random ordering has the longest simulation time.
- Ordering along a hilbert curve has the shortest simulation time, but differences are not significant.
- Further investigations showed that the main time in the simulation is spent on calculating ellipses for the computation of $dist_{ij}$ in the repulsive forces.
- In an optimized program the effects of ordering along a hilbert curve may become more visible.

Mitglied der Helmholtz-Gemeinschaft

JÜLICH FORSCHUNGSZENTRUM

Pedestrian dynamics

Part III: Implementation and analysis of ODE solvers

September 30, 2010 | Timo Hülsmann

Pedestrian dynamics

ODE solvers

- Idea: Increasing step-sizes with more sophisticated solvers
- ODEs for GCFM
- Adaptive Runge-Kutta-Fehlberg Methods
- Velocity Verlet Method
- Test cases
- Results
- Conclusion
- Problems

Idea: Increasing step-sizes with more sophisticated solvers

- A timestep in the GCFM has high computational effort.
- Bigger time-steps could directly decrease the runtime of the simulation.
- Only the explicit Euler-Method was used in the implementation of the GCFM which is of order 1.
- Methods of order > 1 may allow bigger time-steps.

ODEs for GCFM

- System to solve:

$$\frac{d}{dt}\vec{R}_i = \vec{V}_i\,, \quad m_i \frac{d}{dt}\vec{V}_i = \vec{F}_i\,, \quad i = 1, \ldots, N$$

- In the simulation $m_i = 1$ is used, so this can be rewritten as:

$$\vec{x} = (R_{1,x}, R_{1,y}, V_{1,x}, V_{1,y}, \ldots, R_{N,x}, R_{N,y}, V_{N,x}, V_{N,y})^T$$

$$\vec{f}(\vec{x}) = (V_{1,x}, V_{1,y}, F_{1,x}, F_{1,y}, \ldots, V_{N,x}, V_{N,y}, F_{N,x}, F_{N,y})^T$$

$$\frac{d}{dt}\vec{x} = \vec{f}(\vec{x})$$

- We can apply standard numerical integration techniques.

Adaptive Runge-Kutta-Fehlberg 2(3) Method

- We want to integrate the system $\frac{d}{dt}\vec{x} = \vec{f}(\vec{x})$ with step-size h.
- The Runge-Kutta-Fehlberg 2(3) Method reads as follows:

$$\vec{x}_1 = \vec{x}_0 + h\sum_{i=1}^{3} b_i\vec{k}_i\,, \quad \hat{\vec{x}}_1 = \vec{x}_0 + h\sum_{i=1}^{4} \hat{b}_i\vec{k}_i$$

Formula for the increments:

$$\vec{k}_i = \vec{f}(\vec{x}_0 + h\sum_{j=1}^{i-1} a_{ij}\vec{k}_j)\,, \quad i = 1, \ldots, 4$$

Adaptive Runge-Kutta-Fehlberg 2(3) Method

- The coefficients of the method read as follows:

a_{21}=1/4			
a_{31}=-189/800	a_{32}=729/800		
a_{41}=214/891	a_{42}=1/33	a_{43}=650/891	
b_1=214/891	b_2=1/33	b_3=650/891	
$\hat{b}_1$=533/2106	$\hat{b}_2$=0	$\hat{b}_3$=800/1053	$\hat{b}_4$=-1/78

- The local error is estimated by:

$$\vec{e} = \vec{x}_1 - \hat{\vec{x}}_1 = O(3)$$

- Formula for step size prediction:

$$h_{opt} = h_{used} \cdot \sqrt[3]{\frac{1}{\mathrm{ERR}}}$$

Adaptive Runge-Kutta-Fehlberg 2(3) Method

- Error norm:

$$\mathrm{ERR} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(\frac{e_j}{\mathrm{atol} + \max\{|x_{0,j}|, |x_{1,j}|\} \cdot \mathrm{rtol}}\right)^2}$$

- The new step-size is scaled with a safety factor ρ. It must hold:

$$\sigma \cdot h_{used} < h_{new} < \theta \cdot h_{used}$$

- Values of the parameters:

$$\rho = 0.9\,, \quad \sigma = 0.2\,, \quad \theta = 5$$

Adaptive Runge-Kutta-Fehlberg 4(5) Method

- The Runge-Kutta-Fehlberg 4(5) Method uses two methods of order 4 and 5 to get an estimated error of order 5:

$$\vec{x}_1 = \vec{x}_0 + h \sum_{i=1}^{5} b_i \vec{k}_i \,, \quad \hat{\vec{x}}_1 = \vec{x}_0 + h \sum_{i=1}^{6} \hat{b}_i \vec{k}_i$$

with increments:

$$\vec{k}_i = \vec{f}(\vec{x}_0 + h \sum_{j=1}^{i-1} a_{ij} \vec{k}_j) \,, \quad i = 1, \ldots, 6$$

Velocity Verlet Method

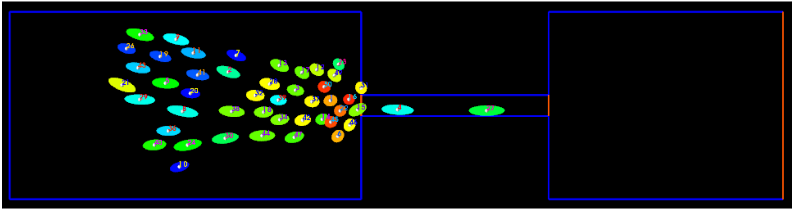
- The Velocity Verlet Method is used in Molecular Dynamics and allows bigger time-steps for dissipative systems.
- The method reads as follows:

$$\vec{R}_i(t+h) = \vec{R}_i(t) + \vec{V}_i(t)h + \frac{1}{2}\vec{F}_i(t)h^2$$

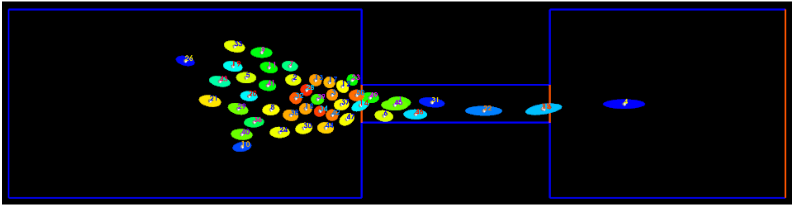
$$\vec{V}_i(t+h) = \vec{V}(t) + \frac{1}{2}\left[\vec{F}_i(t) + \vec{F}_i(t+h)\right] h$$

Test cases

- Bottleneck with 90cm width:



- Bottleneck with 160cm width:



Results - 90cm bottleneck, N=20

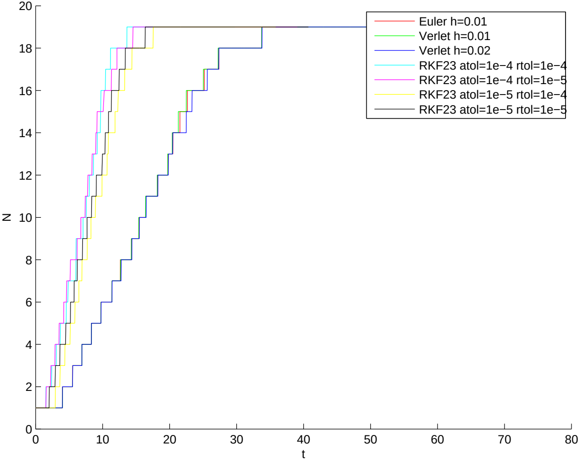


Figure: Pedestrians entered the bottleneck (Euler,Verlet,RKF 2(3))

Results - 90cm bottleneck, N=20

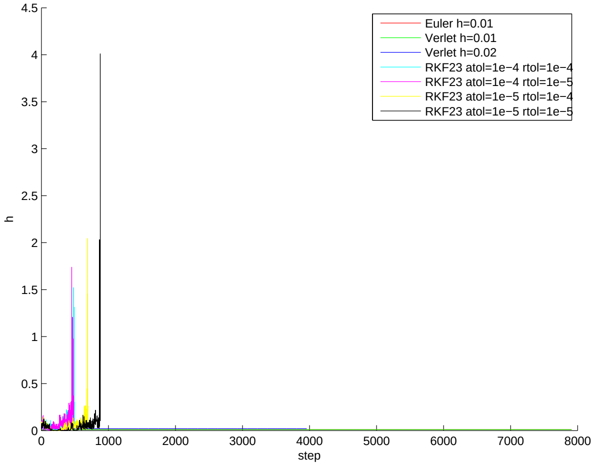


Figure: Step-sizes (Euler,Verlet,RKF 2(3))

Results - 90cm bottleneck, N=20

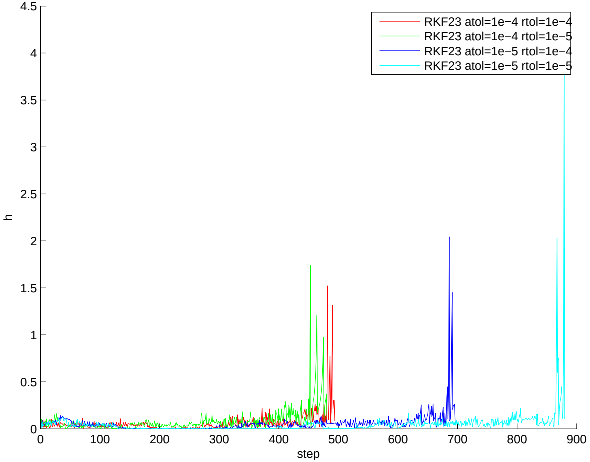


Figure: Step-sizes (RKF 2(3))

Results - 90cm bottleneck, N=20

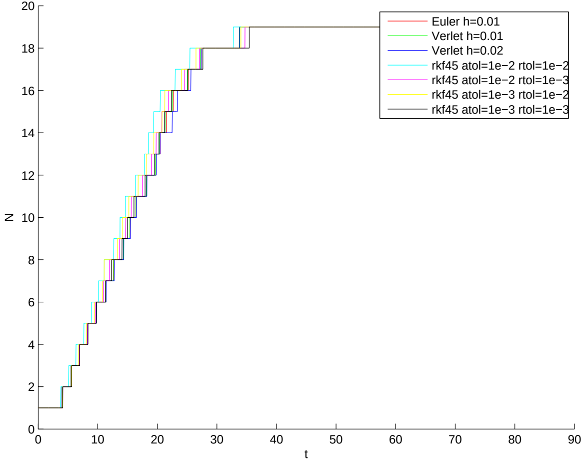


Figure: Pedestrians entered the bottleneck (Euler,Verlet,RKF 4(5))

Results - 90cm bottleneck, N=20

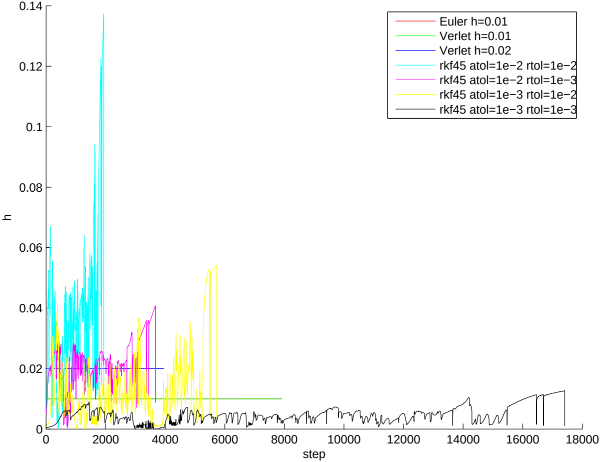


Figure: Step-sizes (Euler,Verlet,RKF 4(5))

Results - 160cm bottleneck, N=50

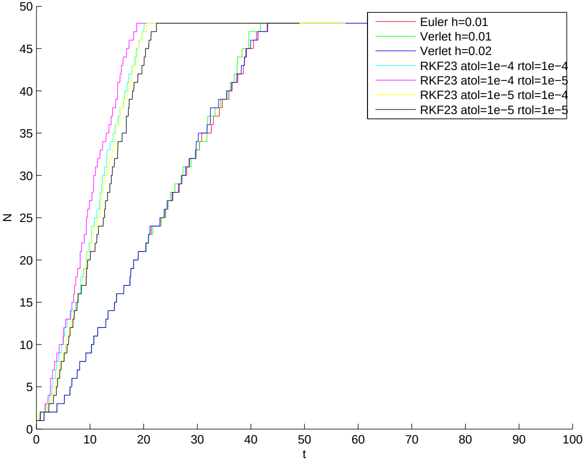


Figure: Pedestrians entered the bottleneck

Results - 160cm bottleneck, N=50

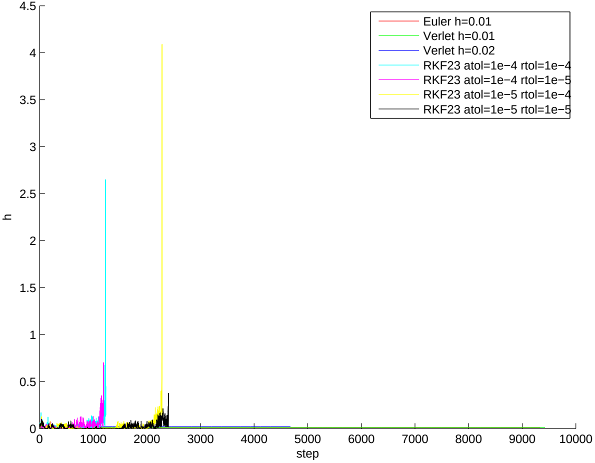


Figure: Step-sizes (Euler,Verlet,RKF 2(3))

Results - 160cm bottleneck, N=50

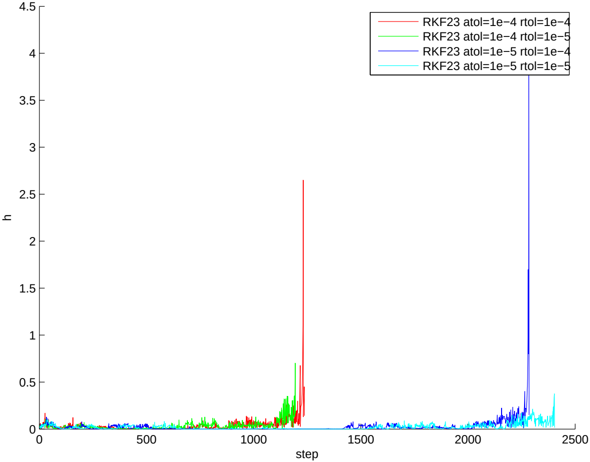


Figure: Step-sizes (RKF 2(3))

Results - 160cm bottleneck, N=50

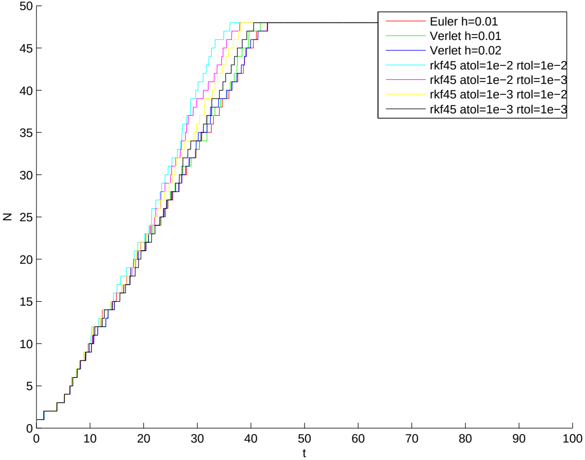


Figure: Pedestrians entered the bottleneck (Euler,Verlet,RKF 4(5))

Results - 160cm bottleneck, N=50

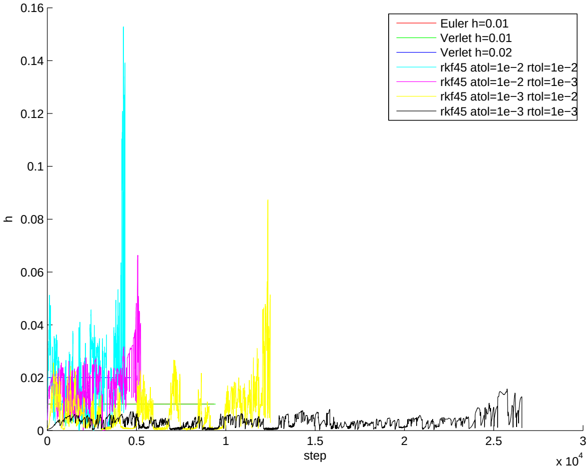


Figure: Step-sizes (Euler,Verlet,RKF 4(5))

Conclusion

- Velocity Verlet allows doubling the step-size and has similar N/t-diagram as Euler-Method.
- Smallest amount of time-steps needed with $atol = 10^{-4}$, $rtol = 10^{-5}$ for RKF 2(3) Method and with $atol = 10^{-2}$, $rtol = 10^{-2}$ for RKF 4(5) Method.
- RKF 2(3) method allows bigger step-sizes but N/t-diagram differs much more from the “reference” N/t-diagram of the Euler-Method than the RKF 4(5) Method.

Problems

- New methods are not stable enough to allow comparison with experimental data (for this simulations with about 180 Pedestrians are needed).
- Some pedestrians are missing in the N/t diagram. This is due to unrealistic increase of forces.

Thank you for your attention!

Author

Timo Hülsmann
University of Wuppertal
42119 Wuppertal, Germany

Disclaimer

Some images used in this talk are intellectual property of other authors and may not be distributed or reused without their explicit approval.