A Deterministic Particle Method for One-Dimensional Reaction-Diffusion Equations

Prof. Dr. Michael Mascagni

Seminar für Angewandte Mathematik, ETH Zürich
Rämistrasse 101, CH-8092 Zürich, **Switzerland**
AND
Department of Computer Science & School of Computational Science
Florida State University, Tallahassee, FL 32306 **USA**

E-mail: `mascagni@cs.fsu.edu` or `mascagni@math.ethz.ch`
URL: `http://www.cs.fsu.edu/~mascagni`

Research supported by ARO, DOE/ASCI, NATO, and NSF

Outline of the Talk

1D Reaction-Diffusion Equations (1DRDEs)

- Consider the reaction-diffusion equation:

$$u_t = \nu u_{xx} + f(u), \quad t > 0, \quad u(x,0) = u_0(x), \quad \nu > 0$$

- Have applications to:
  - Combustion problems
  - Excitable tissue models (nerve, heart, pancreas)
  - Complex chemical reactions
- Various contexts for 1D problems
  - Pure initial value problem: mathematical
  - Initial boundary value problem: modeling

A Monte Carlo method for 1DRDEs

- Consider the above 1DRDE:

$$u_t = \nu u_{xx} + f(u), \quad u(x,0) = u_0(x)$$

- Assume monotonic solution: $u(x,t) < u(y,t)$ when $x < y$ with $u(-\infty,t) = 0$, $u(+\infty,t) = 1$. Then the gradient, $v = u_x$, satisfies:

$$v_t = \nu v_{xx} + f'(u)v, \quad v(x,0) = u_0'(x)$$

A Monte Carlo method for 1DRDEs (cont.)

- With $v(x,t) = \sum_{j=1}^{N} m_j \delta\left(x - X_j(t)\right)$, $X_j(t)$ position and m_j the 'mass' of particle j, we recover as:

$$u(x,t) = \int_{-\infty}^{x} v(x',t)\,dx'$$

$$u(x,t) = \sum_{j=1}^{N} m_j H\left(x - X_j(t)\right)$$

- This is the basis of the 1D random gradient method (RGM) due to Sherman and Peskin with $m_j = \frac{1}{N}$ for $j = 1, 2, \ldots N$

A Monte Carlo method for 1DRDEs (cont.)

- The 1D RGM Algorithm (for each time step):
- Gaussian Random Walk Step: $X_j(t+\Delta t) = X_j(t) + \sigma_j$ where the σ_j are independent $N(0, 2\nu\Delta t)$ random variables
- Evaluate $u_j = u(X_j(t+\Delta t)), j = 1, \ldots, N$ using the above step-function ansatz (equiv. to sorting on $X_j(t)$)
- Kill or Replicate Particles with probability $|f'(u_j)|\Delta t$:
  1. Kill particle if $f' \leq 0$
  2. Rep. particle at X_j if $f' > 0$
- The Ghoniem/Sherman algorithm is similar except that Monte Carlo creation/destruction is replaced by deterministic mass evolution $\frac{dm_j}{dt} = f'(u_j)$

A Deterministic Particle Method for monotonically increasing solutions

- Mark Kac: “use Monte Carlo until you understand the problem"
- Use the function representation for a monotonic solution to the 1DRDE with particle positions defined implicitly by:

$$u(x_i(t),t) = \begin{cases} x_0(t) = -\infty, \\ \frac{i}{N}, \quad i = 1,2,\ldots,N-2,N-1 \\ x_N(t) = +\infty \end{cases}$$

A Deterministic Particle Method (cont.)

- The chain rules gives:

$$\frac{d[u(x_i(t),t)]}{dt} = 0 = u_x(x_i(t),t)\dot{x}_i(t) + u_t(x_i(t),t)$$

$$\text{thus} \quad \dot{x}_i(t) = \frac{-1}{u_x(x_i(t),t)} u_t(x_i(t),t)$$

$$= \frac{-1}{u_x(x_i(t),t)} \Big(\nu u_{xx}(x_i(t),t) + f(u(x_i(t),t)) \Big)$$

- Use formally 2^{nd} order approximations to the various parts of the above: first derivative:

$$\frac{-1}{u_x(x_i(t),t)} \approx -\frac{x_{i+1}(t) - x_{i-1}(t)}{u_{i+1} - u_{i-1}} = -\frac{N}{2}\Big(x_{i+1} - x_{i-1} \Big)$$

A Deterministic Particle Method (cont.)

second derivative:

$$u_{xx}(x_i(t),t) \approx \frac{\frac{u_{i+1}-u_i}{x_{i+1}(t)-x_i(t)} - \frac{u_i-u_{i-1}}{x_i(t)-x_{i-1}(t)}}{\left(\frac{x_{i+1}(t)+x_i(t)}{2}\right) - \left(\frac{x_i(t)+x_{i-1}(t)}{2}\right)}$$

$$= \frac{2}{N\,(x_{i+1}-x_{i-1})}\left[\frac{1}{x_{i+1}-x_i} - \frac{1}{x_i-x_{i-1}}\right]$$

$$= \left[-\frac{N}{2}\left(x_{i+1}-x_{i-1}\right)\right]^{-1}\left[\frac{1}{x_i-x_{i-1}} - \frac{1}{x_{i+1}-x_i}\right]$$

nonlinearity:

$$f(u(x_i(t),t)) = f\left(\frac{i}{N}\right)$$

A Deterministic Particle Method (cont.)

- This leads to the following ODEs:

$$\dot{x}_i = \nu\left[\frac{1}{x_i - x_{i-1}} - \frac{1}{x_{i+1} - x_i}\right] - \frac{N}{2}\left(x_{i+1} - x_{i-1}\right) f\left(\frac{i}{N}\right)$$

- Notice that the PDE diffusion term is nonlinear in the ODE and the PDE nonlinearity is linear in the ODE
- Now consider how to translate PDE boundary conditions for this system of ODEs

Boundary Conditions

- Pure initial value problem: $x_0(t) = -\infty$, $x_N(t) = +\infty$, $\dot{x}_0(t) = \dot{x}_N(t) = 0$:

$$
\begin{aligned}
\dot{x}_1 =& -\nu\left[\frac{1}{x_2 - x_1}\right] \\
& - N\left(x_2 - x_1\right) f\left(\frac{1}{N}\right) \\
\dot{x}_{N-1} =& \nu\left[\frac{1}{x_{N-1} - x_{N-2}}\right] \\
& - N\left(x_{N-1} - x_{N-2}\right) f\left(\frac{N-1}{N}\right)
\end{aligned}
$$

Boundary Conditions (cont.)

- Dirichlet boundary conditions, $u(0,t) = U_0(t)$: use particle creation when $U_0(t)$ is decreasing and diminishes by N^{-1} and particle destruction otherwise
- Neumann boundary conditions, $u_x(0,t) = U_0(t)$: enforced by $x_1(t) = \frac{1}{NU_0(t)}$ with $x_0(t) = 0$

Time Discretization

- Forward Euler:

$$\frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{\Delta t} = F(\mathbf{x}_n) \to \mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t F(\mathbf{x}_n)$$

  is stable if $\frac{\Delta t}{\min_i[(x_{i+1}-x_i)(x_i-x_{i-1})]} \leq \frac{1}{2\nu}$

- Backward Euler:

$$\frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{\Delta t} = F(\mathbf{x}_{n+1}) \to \mathbf{x}_{n+1} - F(\mathbf{x}_{n+1}) = \mathbf{x}_n$$

  leads to a nonlinear system:

$$x_i^{n+1} - \Delta t \nu \left[\frac{1}{x_i^{n+1} - x_{i-1}^{n+1}} - \frac{1}{x_{i+1}^{n+1} - x_i^{n+1}} \right] + \Delta t \frac{N}{2} \left(x_{i+1}^{n+1} - x_{i-1}^{n+1} \right) f\left(\frac{i}{N} \right) = x_i^n$$

Analysis of the Backward Euler Equations

- Define: $K_i^{n+1} = \frac{\nu}{(x_i^{n+1} - x_{i-1}^{n+1})(x_{i+1}^{n+1} - x_i^{n+1})}$ so that the backward Euler equations become:

$$
\begin{aligned}
l_i^{n+1} x_{i-1}^{n+1} + d_i^{n+1} x_i^{n+1} + u_i^{n+1} x_{i+1}^{n+1} &= x_i^n \\
l_i^{n+1} &= -\Delta t \left[K_i^{n+1} + \frac{N}{2} f\left(\frac{i}{N}\right) \right] \\
d_i^{n+1} &= 1 + 2\Delta t K_i^{n+1} \\
u_i^{n+1} &= -\Delta t \left[K_i^{n+1} - \frac{N}{2} f\left(\frac{i}{N}\right) \right]
\end{aligned}
$$

- These equations are (nonlinear) tridiagonal but not diagonally dominant; however a rearrangement produces a diagonally dominant Picard (fixed-point) iteration that is provably convergent

Picard Iteration

- The rearrangement is:

$$\begin{aligned}
L_i^{n+1} x_{i-1}^{n+1} + D_i^{n+1} x_i^{n+1} + U_i^{n+1} x_{i+1}^{n+1} &= \\
x_i^n + \Delta t \frac{N}{2} f\left(\frac{i}{N}\right)\left(x_{i+1}^{n+1} - x_{i-1}^{n+1}\right) \\
L_i^{n+1} &= -\Delta t K_i^{n+1} \\
D_i^{n+1} &= 1 + 2\Delta t K_i^{n+1} \\
U_i^{n+1} &= -\Delta t K_i^{n+1}
\end{aligned}$$

- This system is tridiagonal and diagonally dominant and can be used to define the following Picard iteration (which is linearly convergent) to solve for the new positions at each time-step

Picard Iteration (Cont.)

- The Picard iteration is thus:
  1. Initialization: $\mathbf{x}^{old} = \mathbf{x}^{n}$
  2. Iterate until converged:
     2.1 Solve the following for $\mathbf{x}^{new}$:

$$\begin{aligned} L_i^{old} &= -\Delta t K_i^{old}, \\ D_i^{old} &= 1 + 2\Delta t K_i^{old}, \\ U_i^{old} &= -\Delta t K_i^{old}, \\ L^{old} x_{i-1}^{new} + D^{old} x_i^{new} + U^{old} x_{i+1}^{new} &= \\ x_i^n + \Delta t \frac{N}{2} f\left(\frac{i}{N}\right)\left(x_{i+1}^{old} - x_{i-1}^{old}\right) \end{aligned}$$

     2.2 Set $\mathbf{x}^{old} = \mathbf{x}^{new}$
  3. Form the solution as $\mathbf{x}^{n+1} = \mathbf{x}^{new}$

Newton Iteration

- Newton iteration should be quadratically convergent, and is based on the solving the nonlinear equation:

$$
\begin{aligned}
G_i(\mathbf{x}) = x_i - \nu \Delta t \left[\frac{1}{x_i - x_{i-1}} - \frac{1}{x_{i+1} - x_i} \right] \\
+ \Delta t \frac{N}{2} (x_{i+1} - x_{i-1}) f\left(\frac{i}{N} \right) - x_i^n
\end{aligned}
$$

Newton Iteration (cont.)

- This is solved as $\mathbf{J}(\mathbf{x}_{old})(\mathbf{x}_{new} - \mathbf{x}_{old}) = -\mathbf{G}(\mathbf{x}_{old})$ where the Jacobian is defined as:

$$[\mathbf{J}]_{ij} = \begin{cases} \frac{-\nu\Delta t}{(x_{i+1}-x_i)^2} + \frac{N\Delta t}{2} f\left(\frac{i}{N}\right) & j = i+1 \\ 1 + \left[\frac{\nu\Delta t}{(x_{i+1}-x_i)^2} + \frac{\nu\Delta t}{(x_i-x_{i-1})^2}\right] & j = i \\ \frac{-\nu\Delta t}{(x_i-x_{i-1})^2} - \frac{N\Delta t}{2} f\left(\frac{i}{N}\right) & j = i-1 \\ 0 & \text{otherwise} \end{cases}$$

- $[\mathbf{J}]_{ij}$ is tridiagonal but not symmetric and not, in general, diagonally dominant unless $[\mathbf{J}]_{ii+1} < 0$

Newton Iteration (cont.)

- This is equivalent to $\frac{\nu \Delta t}{(x_{i+1}-x_i)^2} > \frac{N\Delta t}{2} f\left(\frac{i}{N}\right)$
- Define $f_{max} = \max_{x\in[0,1]} f(x)$, and divide by $\Delta t N^2$ to give $\frac{\nu}{[N(x_{i+1}-x_i)]^2} > \frac{1}{2N} f_{max}$
- Now $\frac{1}{[N(x_{i+1}-x_i)]^2} \approx u_x(x_i(\cdot),\cdot)^2 > H^{-1} > 0$ by monotonicity
- Rewriting (heuristically) $N > \frac{Hf_{max}}{2\nu}$, since the right-hand side is constant, choosing N larger makes the inequality true and the system diagonally dominant

Numerical Results

- Consider the concrete 1DRDE, Nagumo's equation (a model of nerve conduction):

$$u_t = u_{xx} + u(1-u)(u-a), \quad t > 0,$$
$$u(x,0) = u_0(x), \quad 1 \geq a \geq 0$$

- Good for numerical experimentation because:
  - Exact solution $u(x,t) = \frac{1}{1+e^{-(x-\theta t)/\sqrt{2}}}$
  - A traveling wave with wave speed $\theta = \sqrt{2}\left(a - \frac{1}{2}\right)$

Numerical Results (cont.)

- With $a = \frac{1}{2}$ we have $\theta = 0$ and the solution is a stable standing wave ideal for studying convergence (in N) of the method
- Empirical comparison of Picard versus Newton iterations for the backward Euler equations:
  - Usually fewer Picard iterations required
  - Often Picard iterations increased and convergence ceased while Newton settled into a constant number of iterations

Numerical Results (cont.)

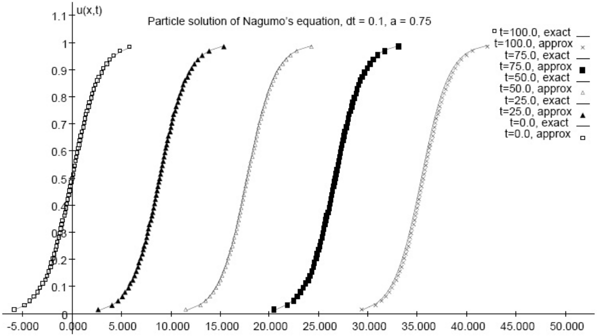

Figure: Solution to Nagumo's equation with $a = 0.75$, $\Delta t = 0.1$ with $N = 64$ particles, printed at $t = 0.0, 25.0, 50.0, 75.0$, and 100.0.

Numerical Wave Speed Results

- Note that the particles lead the solution (slightly)
- The exact midpoint wave speed is

$$\dot{x}_{mid}(t) = \frac{-1}{u_x(x_{mid}(t), t)} \left[u_{xx}(x_{mid}(t), t) + \frac{1}{2} \left(1 - \frac{1}{2}\right) \left(\frac{1}{2} - a\right) \right]$$

- $u_x(x_{mid}(t), t) = \frac{1}{4\sqrt{2}}$
- $u_{xx}(x_{mid}(t), t) = 0$
- $\dot{x}_{mid}(t) = -4\sqrt{2} \left[\frac{1}{4} \left(\frac{1}{2} - a \right) \right] = \sqrt{2} \left(a - \frac{1}{2} \right) = \theta$

Numerical Wave Speed Results (cont.)

- The numerical wave speed of the midpoint is

$$\dot{x}_{mid} = \nu \left[\frac{1}{x_{mid} - x_{mid-1}} - \frac{1}{x_{mid+1} - x_{mid}} \right] - \frac{N}{2}\left(x_{mid+1} - x_{mid-1} \right) f\left(\frac{i}{N} \right)$$

- Can prove: $(x_{mid+1} - x_{mid}) = (x_{mid} - x_{mid-1})$
- We use: $\frac{N}{2}(x_{mid+1} - x_{mid-1}) = \frac{N}{2} 2h \approx \frac{1}{u_x(x_{mid},t)}$

Numerical Wave Speed Results (cont.)

- Can prove:

$$
\begin{aligned}
\frac{\frac{2}{N}}{2h} &= \frac{u(x_{mid}+h,t)-u(x_{mid}-h,t)}{2h} \\
&= u_x(x_{mid},t) + \frac{h^2}{6}u_{xxx}(x_{mid},t) + O(h^4) \rightarrow \\
\frac{N}{2}2h &= \frac{1}{u_x(x_{mid},t)+\frac{h^2}{6}u_{xxx}(x_{mid},t)+O(h^4)} \\
&= \frac{1}{u_x(x_{mid},t)}\left[1-\frac{h^2}{6}\frac{u_{xxx}(x_{mid},t)}{u_x(x_{mid},t)}+O(h^4)\right]
\end{aligned}
$$

Numerical Wave Speed Results (cont.)

- $u_{xxx}(x_{mid}, t) = -\frac{1}{16\sqrt{2}}$
- This gives:

$$\begin{aligned} \dot{x}_{mid}(t) &= \frac{-f\left(\frac{1}{2}\right)}{u_x(x_{mid}(t), t)}\left[1 + \frac{h^2}{24} + O(h^4)\right] \\ &= \theta\left[1 + \frac{h^2}{24} + O(h^4)\right] \end{aligned}$$

- Now we also have a proof using N instead of h

Numerical Wave Speed Results (cont.)

$N \downarrow a \rightarrow$	0.125	0.25	0.375	0.5
4	-1.71e-01	-1.33e-01	-7.32e-02	0.0e+00
8	-1.04e-01	-7.36e-02	-3.77e-02	0.0e+00
16	-4.70e-02	-2.68e-02	-1.18e-02	0.0e+00
32	-1.91e-02	-8.78e-03	-3.23e-03	0.0e+00
64	-7.62e-03	-2.80e-03	-8.23e-04	0.0e+00
128	-3.06e-03	-9.00e-04	-1.96e-04	0.0e+00
256	-1.25e-03	-2.91e-04	-4.35e-05	0.0e+00
512	-5.14e-04	-9.58e-05	-8.72e-06	0.0e+00
1024	-2.13e-04	-3.19e-05	-1.43e-06	0.0e+00
2048	-8.88e-05	-1.07e-05	-1.15e-07	0.0e+00
4096	-3.71e-05	-3.68e-06	4.73e-08	0.0e+00

The error in the wave speed (θ) in the particle method solution to Nagumo's equation: errors for various values of a and N are presented

Numerical Error Results

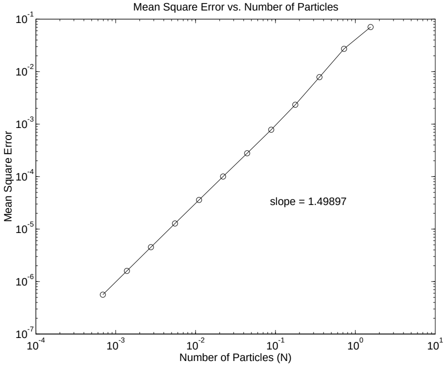

Figure: Mean square error of the particle method solution for various number of particles; the solution was computed with $a = \frac{1}{2}$ giving a wave speed of $\theta = 0$

Numerical Error Results (cont.)

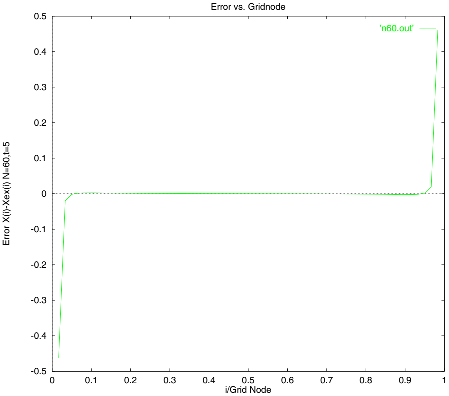

Figure: Error as a function grid point; the solution was computed with $a = \frac{1}{2}$ giving a wavespeed of $\theta = 0$

Numerical Error Results (cont.)

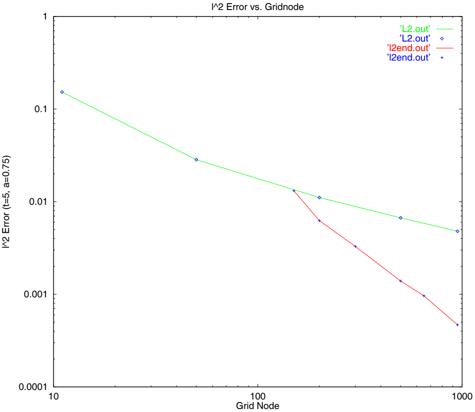

Figure: Mean square error omitting the boundary points; the solution was computed with $a = \frac{1}{2}$ giving a wavespeed of $\theta = 0$

Getting Second Order Accuracy

- The problem is clearly at the end-points, so how can we fix them up?
  - With $x(u,t)$, have singularities at $u=0$ and $u=1$
  - Can consider approximating $x = a/u + b$ near the singularities, so that $x_i = a/u_i + b$, $i = 1, 2$
  - Need to compute $\frac{\partial x}{\partial u}$ at u_1 to compute the correction, this is $2N(x_2 - x_1)$
  - Leads to the following "corrected" boundary terms:

$$\dot{x}_1 = -\nu \left[\frac{1}{x_2 - x_1} \right] - 2N\,(x_2 - x_1)\, f\left(\frac{1}{N} \right)$$

$$\dot{x}_{N-1} = \nu \left[\frac{1}{x_{N-1} - x_{N-2}} \right] - 2N\,(x_{N-1} - x_{N-2})\, f\left(\frac{N-1}{N} \right)$$

- When using this, one obtains an empirical $N^{-1.92}$ convergence behavior!

Conclusions

- Have a deterministic particle method for reaction diffusion equations
  - Discretization of the solution
  - Naturally adaptive
  - Good for steep gradients
- Analyzed forward and backward Euler methods
  - Forward Euler has usual stability requirement
  - Backward Euler has Picard and Newton
- Have proof that particles cannot cross
- Have computed solutions to Nagumo's equation:
  - Wave speed discrepancy understood
  - Have computed $O(N^{-2})$ convergence far from the endpoints

Open Problems

- How do we improve the boundary conditions to uniformly get $O(N^{-2})$ convergence? (Solved!)
  - Better boundary conditions?
  - More refinement near points at infinity?
- The infamous *Sign Problem* for nonmonotonic solutions
  - Using positive and negative particles leads to cancelation
  - Can make policies for Monte Carlo and deterministic
- Systems, branching geometry
- Higher spatial dimensions

References I

A. F. Ghoniem and F. Sherman (1985)
Grid-free simulation of diffusion using random walk methods
Journal of Computational Physics, **61**: 1–37.

A. L. Hodgkin and A. F. Huxley (1952)
A quantitative description of membrane current and its application to conduction in the giant axon of *L*oligo
Journal of Physiology, **117**: 500–544.

M. Mascagni (1990)
The backward Euler method for the numerical solution of the Hodgkin-Huxley equations of nerve conduction
SIAM Journal on Numerical Analysis, **27**: 941–962.

M. Mascagni (1995)
A deterministic particle method for one-dimensional reaction-diffusion equations
Research Institute for Advanced Computer Science (RIACS) Technical Report: 95.23, *Institute for Defense Analyses/Center for Computing Sciences (IDA/CCS) Technical Report*: CCS-TR-95-144.

J. Nagumo, S. Arimoto and S. Yoshizawa (1962)
An active pulse transmission line simulating nerve axon
Proceedings of the Institute of Radio Engineers, **50**: 2061–2070.

References II

G. Russo (1990)
Deterministic diffusion of particles
Communications on Pure and Applied Mathematics, **63**: 697–733.

A. S. Sherman and C. S. Peskin (1986)
A Monte-Carlo method for scalar reaction diffusion equations
SIAM Journal on Scientific and Statistical Computing, **7**: 1360–1372.

A. S. Sherman and M. Mascagni (1994)
A gradient random walk method for two-dimensional reaction-diffusion equations
SIAM Journal on Scientific Computing, **15**: 1280–1293.