

Analysis of a Parallel Solver for the Gray-Scott Reaction-Diffusion System

Parallel Programming for Large-Scale Problems SF2568
Teacher: Michael Hanke

Jonathan Ridenour, 780514-7779
Mateusz Herczka, 700624-9234

May 2, 2015

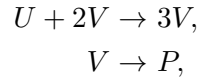
Problem description

The subject of this report is the parallelization of a numerical method for a two-component reaction-diffusion system known as the Gray-Scott model. Originally introduced by Gray and Scott [1], the model is a system of two partial differential equations for the concentration of two chemical species, which are reacting with each other as they diffuse through a medium.

Pattern formation resulting from the Gray-Scott model is a rich area of research, with applications to a variety of physical, chemical, and biological phenomena, as well as cellular automata and the pattern formation of other partial differential equation systems [2]. We investigate a parallelization strategy for the two-dimensional case. We make a series of trials, running the solver in parallel on 4, 16, 36, and 64 processes, and perform a speedup analysis using the results of these trials. Finally, we show a few of the various patterns that emerge from the Gray-Scott model with various input parameters.

Mathematical Formulation

The Gray-Scott model involves the reaction and diffusion of two generic chemical species, U and V , whose concentrations are described by the functions $u(x, y, t)$ and $v(x, y, t)$, reacting according to the chemical equations



where P is an inert byproduct. This system is governed by the following system of partial differential equations, known as the Gray-Scott equations:

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u - uv^2 + F(1 - u), \quad (1)$$

$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + uv^2 - (F + K)v. \quad (2)$$

Here, D_u and D_v are the diffusion constants for u and v respectively, and F and K are constants which govern the replenishment of the chemical species.

Numerical Method

In order to discretize (1) and (2) we let....

...thus we can write the fully discretized equations as

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t[A\mathbf{u}^n + f_u(\mathbf{u}^n, \mathbf{v}^n)], \quad (3)$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \Delta t[A\mathbf{v}^n + f_v(\mathbf{u}^n, \mathbf{v}^n)], \quad (4)$$

and we have an exact update formula for each time step.

Stability Conditions

In order to ensure numerical stability for the differential operator A in (3), we must conform to the restriction

$$\Delta t \lambda_k \in \mathcal{S}, \forall k, k = 1, 2, \dots, N,$$

where λ_k is the k -th eigenvalue of A , and \mathcal{S} is the stability region of the forward Euler method: a circle in the complex plane centered at -1 with unit radius [3]. The maximum eigenvalue, λ_{max} for the chosen parameters has a strictly negative real value, and thus we must choose a timestep such that $\Delta t \cdot \lambda_{max}$ is at least -2 . The maximum eigenvalue is computed using Matlab as:

$$\lambda_{max} = -2.547,$$

which provides the limitation on the timestep:

$$\Delta t \leq \frac{-2}{-2.547} = 0.7852.$$

Since the timestep also has a damping effect on the replenishment term and reaction rate, we must consider these as well when choosing the timestep. We settle on

$$\Delta t = 0.25.$$

This is well within the stability bounds of A , and provides suitable damping on the replenishment and reaction rate terms for Gray-Scott pattern formation as studied in [2].

Algorithm Description

Implementation details

The computational domain is divided into subdomains, one for each process. Each process is responsible for the update of its local values. Along the subdomain boundaries, neighbouring values are needed for the update; these are obtained by coordinated message passing. The boundary data are communicated between processes by means of a red-black communication schedule. When all necessary information is obtained, the local values are updated according to (3).

Pseudocode

```
MPI Initialize;  
Calculate mesh position and neighbouring ranks;  
Determine color according to rank;  
Set initial values according to rank;  
for Max iterations:
```

```
Build boundary output data;
if (red):
    for each neighbour:
        MPI send boundary data;
    end
    for each neighbour:
        MPI receive boundary data;
    end
else:
    for each neighbour:
        MPI receive boundary data;
    end
    for each neighbour:
        MPI send boundary data;
    end
end
Update values;
end
Print results;
MPI Finalize;
```

Theoretical performance evaluation

Testing with different cases

Experimental speedup investigation with meaningful data

References

- [1] Gray P, Scott SK. *Sustained oscillations and other exotic patterns of behavior in isothermal reactions*. J Phys Chem 1985;89:22–32.
- [2] Weiming Wang, Yezhi Lin, Feng Yang, Lei Zhang, Yongji Tan, *Numerical study of pattern formation in an extended Gray–Scott model*, Communications in Nonlinear Science and Numerical Simulation, 16 (2011).
- [3] Lennart Edsberg *Introduction to Computation and Modelling for Differential Equations*, 2008, John Wiley and Sons, pp 50.