Primordial Particle System

An analysis by ibra-kdbra

In 2015, Thomas Schmickl and Martin Stefanec of the University of Graz, Austria, released a paper detailing a computerized particle system capable of spontaneously generating cell-like patterns and behaviours. Much like Conway's Game of Life, Schmickl and Stefanec's Primordial Particle System uses very simple rules to produce complex behaviour. However, unlike Game of Life the Primordial Particle System has several parameters that may be tuned, leading to different kinds of 'life' forming (or not forming).

In their original paper, Schmickl and Stefanec fully explored the two-dimensional search space of alpha and beta values. Having found an interesting 'lifeform' using the values $\alpha=180°$, $\beta=17°$, $r=5$, $v=0.67$, their search through alpha and beta values used the same fixed values of r and v.

Having recreated their setup using JavaScript I have explored many different values of α, β, r and v. Here I will describe some of my findings, and some suggestions for further exploring the search space.

Species Parameters

First of all let's consider the value of s, the visible size of particles. In the original paper r and v are in units proportional to s; $r=5$ means $r=5s$ and $v=0.67$ means $v=0.67s$. However, since the size of particles does not affect the mathematics and particles are essentially point-like, using the value of s is as arbitrary as measuring in inches rather than centimetres.

In order to reduce the number of free parameters, making the search space easier to explore while also making the simulations scale invariant, I have done away with units altogether. Instead, the step size of particles (the distance they move during each iteration) can be described using the ratio v/r, which I call 'gamma' (γ), a unitless value.

It can be readily demonstrated that specific combinations of alpha, beta and gamma produce particles with a specific behaviour that is invariant with scale, which is to say doubling or halving both r and v will have no effect on behaviour, just as measuring r in centimetres rather than inches has no effect. For this reason I will refer to alpha, beta and gamma as species parameters, and any specific values of alpha, beta and gamma I will consider as a specific species.

Exploring this three-dimensional search space I have already discovered several interesting species, some of which exhibit cell-like behaviours of growth and division.

Environmental Controls

Since a species' behaviour is affected by local particle density (ie, number of neighbours N) it can be readily demonstrated that in a finite system, where there is only so much area in which to spread, the overall average density also has an effect. Since many species exhibit diffusion at low densities (average N tends towards zero), controlling overall average density (d) allows you to force particles together and increases the probability of creating higher local densities here and there, which often act as nucleation sites for more interesting behaviours.

In essence, different species require different particle densities before exhibiting cell-like structures and behaviours (if at all). For example, the species described in the original paper (α180°, β17°, γ0.134) requires a minimum density of around 800% before forming cell-like aggregations. In a finite system, setting the density to d=8 almost guarantees cell formation. In an unbounded system, setting the initial density to d=8 has much the same effect, although d will slowly reduce over time as particles diffuse into the available space.

It's also fairly straightforward to create an initial spread of particles that is centre-weighted, such that a range of local densities is created at the outset. In such a simulation a species will exhibit a full range of behaviour. Particles on the periphery tend to show only their intrinsic turning behaviour (α), while particles near the centre usually form persistent, dense aggregates.

Many species tend to diffuse at low densities, aggregate at higher densities, diffuse at even higher densities, and aggregate at yet higher densities and so on. This is the reason my many species grow and divide like cells; particles are effectively attracted to each other at a given density (N), but begin to repel each other once a certain threshold is reached.

The rotational behaviour, which depends on the sign of alpha (positive means clockwise, negative means anticlockwise), also usually flips as density increases, such that you often see internal contrarotation, with the particles on the outside of a group cycling in one direction while the particles on the inside, at higher local densities (N), cycle in the opposite direction.

Optimization

By choosing values of alpha, beta and gamma, and controlling density (either by using a centre-weighted distribution or by varying d values) the simulation also becomes invariant to particle number. A simulation using 200 particles will produce much the same structures as a 10,000 particle simulation, the only differences being caused by edge effects, since the particles can pass from one edge of the screen and appear on the opposite edge but are not affected by their would-be neighbours at the other edge due to the extra load that that would place on the processor.

The practical upshot of this is that many low particle number simulations can be performed very quickly, and once an interesting species is discovered you can increase the particle number to reduce edge effects, rather than slow everything down by rewriting the code to do away with edge effects altogether.

It can also be readily demonstrated that all species with positive values of alpha are identical in behaviour to their negative alpha counterparts but are simply mirror opposites, therefore negative values of alpha can be ignored and the search space effectively halved.

Another interesting observation is that affecting all 3 species parameters simultaneously (eg, halving all 3 values) results in a proverbial 'cousin' of the original species, where the only qualitative differences are due to the granularity of time. With enough processing power it may be possible to run simulations as more of a continuous process, although the interesting effects of using a stepwise process may be lost. What seems clear is that different 'lifeforms' do not just inhabit a single point in the 3D search space but are found throughout diagonal swathes of this search space, within limits.

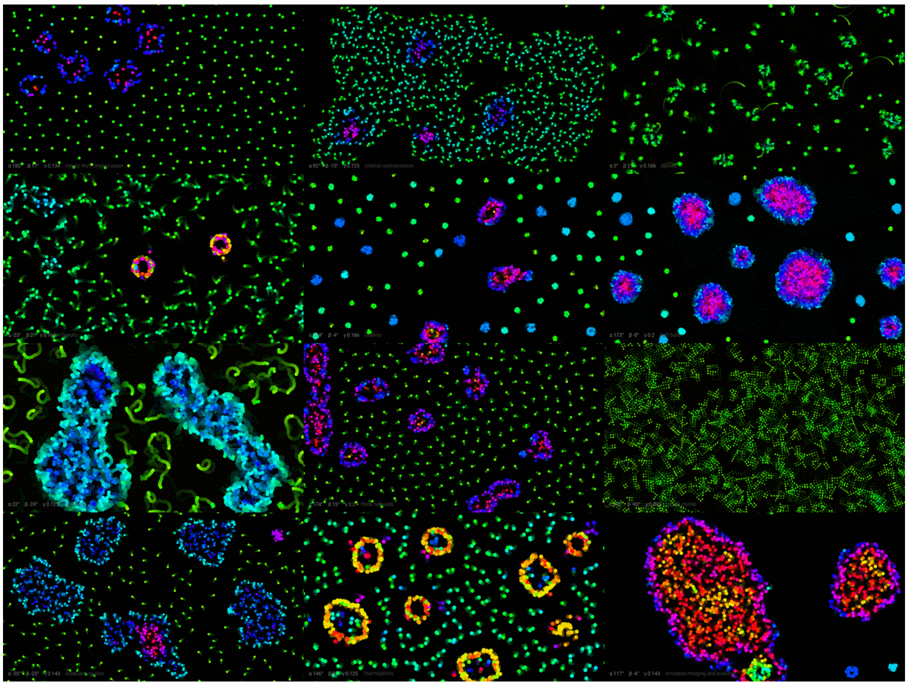
Conclusion

In conclusion, I recommend that any future automated searches focus on values of alpha, beta and gamma, where alpha is always positive, and use overall average density (d) as a metric, both to control and assess simulations. Varying values of r and v, or using negative values of alpha, will result in a duplication of effort, wasting valuable processor cycles.

The latest version of my JavaScript implementation can be used to manually explore the reduced search space in the manner described, or easily modified to perform automated searches. The single-page .htm file, which uses inline script and CSS, can be found here:

https://drive.google.com/open?id=1X_ySTiIS6T46CPZrzc7PSOO3xW70wQe0

No license is required and everyone is free use or modify the code however they see fit, although giving me a mention as the original author as well as crediting the authors of the original paper is advised.



Snapshots of a dozen different species

Notes

Overall average density (d) can be fixed using the size of the HTML canvas or Viewport and the total number of particles (n), thus:

Where w is the total available width and h is the total available height,

The area available to each particle: $(w \times h) / n$,

The area of the neighbourhood around each particle: $\pi \times r^2$,

Therefore, for a given radius r: $d = (\pi \times r^2 \times n) / (w \times h)$,

Conversely, for fixed values of d: $r = \sqrt{(w \times h \times d) / (\pi \times n)}$