37.5 ▪ Problems to Solve

37.5.1 ▪ The Complex Ginsburg–Landau Equation

The complex Ginsburg–Landau equation can exhibit pattern formation for certain values of the coefficients [30]. The equation is given by

$$\frac{\partial A}{\partial t} = A + (1 + ic_1)\nabla^2 A - (1 - ic_3)|A|^2 A, \tag{37.14}$$

where $A(\mathbf{x}, t)$ is a complex valued field.

Assignment. Solve the complex Ginsburg–Landau equation in two dimensions (or three dimensions) on a domain that is $L = 128\pi$ on each side. Converting to a length of 2π on each side means that Equation (37.14) is rescaled to be

$$\frac{\partial A}{\partial t} = A + \left(\frac{2\pi}{L}\right)^2 (1 + ic_1)\nabla^2 A - (1 - ic_3)|A|^2 A. \tag{37.15}$$

Use the pseudospectral method with the fourth-order Runge–Kutta method using random initial data in the range $[-1.5, 1.5] + i[-1.5, 1.5]$ and run until terminal time $T = 10^4$. To see any resulting patterns, use a contour plot on the amplitude $|A|$ and also a contour plot on the phase using `atan2(A.imag,A.real)`. For the parameters $c_1 = 1.5, 0 < c_3 \leq 0.75$, spiral waves should emerge.

Your program should take four arguments with an optional fifth argument for the seed value so that your program named `cgl` can be executed like this:

```
$ cgl 128 1.5 0.25 100000
```

where $N = 128$ is the size of the grid in both x- and y-dimensions, $c_1 = 1.5$, $c_3 = 0.25$ are the real-valued coefficients in Equation (37.15), and $M = 100{,}000$ is the number of time steps. If a fifth argument is specified, then it is a long integer seed value, s, for generating the initial random values. You will need to check the value of `argc` to determine if s was given. Your program must print the arguments given including the value of the seed, and you should verify that your seed argument is implemented correctly by taking the value printed by your program from the above test run and adding it to the argument list. For example, if your program reports using a seed of 12345, then execute your program again like

```
$ cgl 128 1.5 0.25 100000 12345
```

and verify that you get the exact same results.

Note that because this is on a periodic domain, the space step size will be $\Delta x = \Delta y = \frac{2\pi}{N}$ with $x_0 = -\pi$ and $x_{N-1} = \pi - \Delta x$. Your program should output the grid values for A in a file called "`CGL.out`" that contains *only* the data at the time points $t = 1000k$ for $k = 0, 1, \ldots, 10$.

Your program must also print to the screen the total time required to complete the calculation. Use varying values of N to generate a plot that illustrates the scaling of the computational cost versus N.

Figure 37.1 shows an example of the solution at the terminal time $T = 10{,}000$ for the case of $L = 128\pi$, $c_1 = 1.5$, and $c_3 = 0.25$.

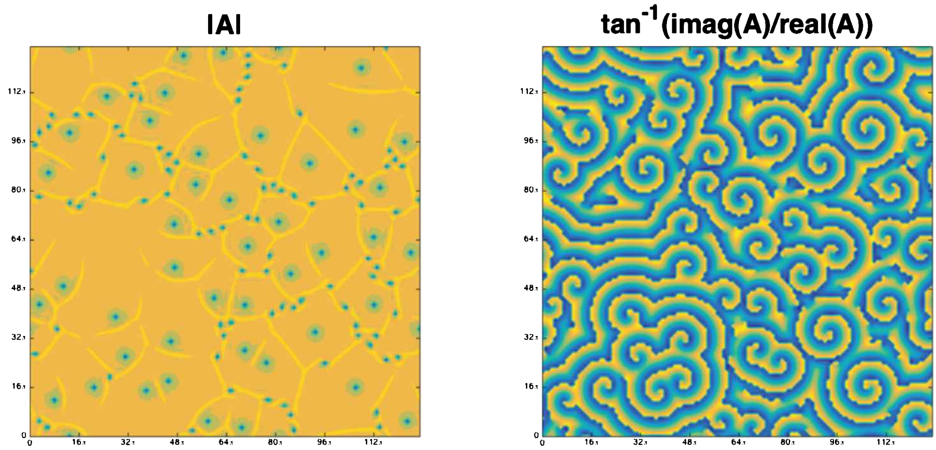

Figure 37.1. *Illustration of a sample solution at the terminal time* $T = 10{,}000$ *for the case of* $c_1 = 1.5$, $c_3 = 0.25$, *and* $L = 128\pi$ *in Equation (37.15). On the left is a contour plot of the magnitude* $|A|$ *and on the right is a contour plot of the phase,* $\arg(A) = \tan^{-1}(Im(A)/Re(A))$. *The spiral patterns that emerge can be seen in the right-hand plot. Results will vary according to the random initial data.*

37.5.2 ▪ Allen–Cahn Equation

The Allen–Cahn equation [31] is an equation that can be used to study phase separation in multicomponent alloys, among other things. It is a balance between a diffusion operator that smooths the transition area between the phases and the free energy density that drives the material to separate into one phase where $\phi = -1$ or the other phase where $\phi = 1$. It is a general equation, but for this example it will take the form

$$\frac{\partial \phi}{\partial t} = -\mathbf{v} \cdot \nabla \phi + b \left(\nabla^2 \phi + \frac{\phi(1-\phi^2)}{W^2} \right). \tag{37.16}$$

Use the pseudospectral method with the fourth-order Runge–Kutta method using random initial data in the range $-1 \le \phi \le 1$ and run until the terminal time $T = 5$. Use $b = 0.25$, $W = 0.25$, and $\mathbf{v} = [10, -5]$ in two dimensions or $\mathbf{v} = [-10, -5, 0]$ in three dimensions.

Your program should take six arguments with an optional seventh argument for the seed value so that your program named `allen` can be executed like this:

```
$ allen 128 10 -5 0.25 0.25 50000
```

where $N = 128$ is the size of the grid in both x and y dimensions, $\mathbf{v} = (10, -5)$ are the two components of the velocity vector, $b = 0.25$ and $W = 0.25$ are the values of the coefficients, and $M = 50{,}000$ is the number of time steps to use. If a seventh argument is specified, then it is a long integer seed value, s, for generating the initial random values. You will need to check the value of `argc` to determine if s was given. Your program must print the arguments given including the value of the seed, and you should verify that your seed argument is implemented correctly by taking the value printed by your program from the above test run and adding it to the argument list. For example,