SOLUTIONS TO ASSIGNMENT 4

Nick Trefethen, 5 March 2019

These solutions are written with Matlab Publish — please excuse imperfect formatting. As you'll see below, the six numbers are about 1.06036, 3.85221, 3437.5, 6.0589, 3.90005, and 0.0281. Also, please excuse the extensive use of Chebfun. I reached for the tool I know best, and there is no expectation that you had to use Chebfun too.

This was a big assignment. The whole course through two terms was a major effort, and congratulations on reaching the end! I have been most impressed with people's hard work and creativity, and I am sorry it hasn't been possible to get to know most of you in person.

1. Quartic Schrodinger equation

This problem is readily solved with Chebyshev spectral methods. We can modify the M-files given in the course or do it with Chebfun, like this:

```
L = chebop(-5,5);
L.op = @(x,u) -diff(u,2) + x.^4.*u;
L.bc = 'dirichlet';
e = eigs(L)

e = 1.06036209
    3.79967303
    7.45569794
   11.64474551
   16.26182602
   21.23837292
```

This suggests that the answer is $1.060362\ldots$. Is that accurate, and is it the right eigenvalue? Well it's probably the right eigenvalue, since the quadratic Schrödinger operator has have eigenvalues $1, 3, 5, 7, \ldots$ on the infinite interval. On $[-5, 5]$ the quadratic operator gives via Chebfun

```
L.op = @(x,u) -diff(u,2) + x.^2.*u; e = eigs(L)

e = 1.000000000
    3.000000007
    5.000000168
    7.000002443
    9.000025175
   11.000197436
```

Since x^4 looks roughly like x^2 — both potential wells will confine this low eigenvalue to a region approximately $[-1, 1]$ — it looks as if $1.06036\ldots$ should indeed be the right eigenvalue. The fact that the quadratic analogue matches 1 to 9 digits suggests it's very likely that $1.06036\ldots$ is accurate to a number of digits. We could compute the residual like this:

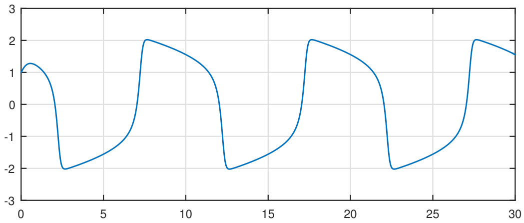
```
[v,lam] = eigs(L,1);
norm(L*v-lam*v)

ans = 5.2587e-12
```

which further suggests that there are quite a few digits of accuracy.

2. Van der Pol equation

The critical value is $a \approx 3.85221$. We could explore this with the code m26 from lectures, or in Chebfun, like this:

```
a = 3.85221;
N = chebop(@(u) diff(u,2) - a*(1-u^2)*diff(u) + u,[0 30]);
N.lbc = [1; 1]; u = N\0; plot(u)
```



The corresponding period is evidently about 9.9999993.

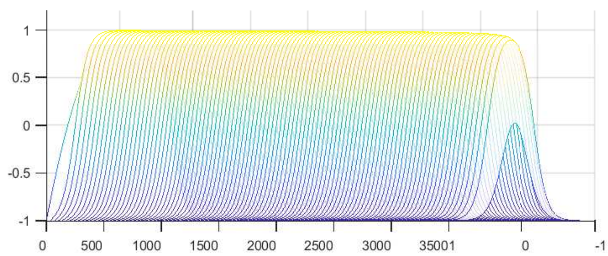
```
[val,pos] = max(u,'local'); diff(pos)

ans =
    7.0900136
    9.9999993
    9.9999993
```

How did I find 3.85221? I just did it by hand using bisection. Of course automated approaches can be used. For example, if a is increased to 3.852211, the period goes up to 10.0000007.

3. Allen-Cahn equation

I used Chebfun's pde15s code for this one, again running bisection by hand. The critical time emerges as about 3437.5, as this computation shows.

```
t = 0:34.375:3437.51;
pdefun = @(t,x,u) .015*diff(u,2)+u-u.^3;
bc.left = @(t,u) u+1; bc.right = @(t,u) u+1;
x = chebfun(@(x) x); u0 = 1-2.*x.^2;
opts = pdeset('Eps', 1e-6, 'Ylim', [-1,1.1]);
[t, u] = pde15s(pdefun, t, u0, bc, opts);
waterfall(u,t), view(110,0), zlim([-1 1.2])
```



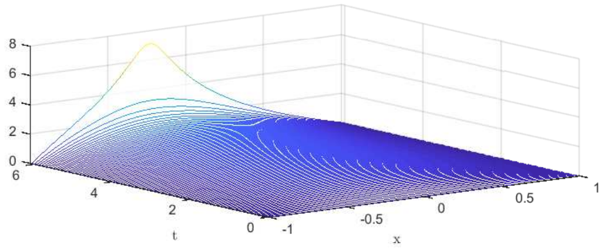
4. Blowup problem

Using codes from the lectures, I would probably adapt m58.m for this problem. In Chebfun, I used `pde15s` again.

I think the critical time is about 6.0589. This is based on the following kind of computation. I varied the value of Eps and the number of time steps and got consistent results.

```
t = linspace(0,6.058,100);
pdefun = @(t,x,u) diff(u,2)+diff(u)+exp(u);
bc.left = 'dirichlet'; bc.right = 'dirichlet';
x = chebfun(@(x) x); u0 = chebfun(0);
opts = pdeset('Eps', 1e-7);
[t, u] = pde15s(pdefun, t, u0, bc, opts);
waterfall(u, t), xlabel('x'), ylabel('t')
max(u(:,end)), axis([-1 1 0 6.1 0 8])
```

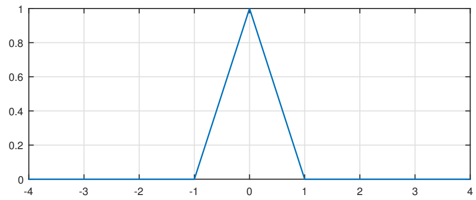
```
ans = 7.1241819
```



5. Advection-diffusion equation

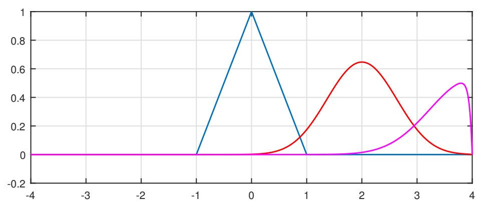
The solution to this problem is a pulse that travels to the boundary at the right, then decays. We can simulate it with one of the codes from the course and then measure the maximum in various ways. We can also do it like this with Chebfun. First we set up the initial condition and differential operator:

```
x = chebfun('x',[-4 4]); u0 = max(0,1-abs(x)); L = chebop(-4,4);
L.op = @(u) diff(u,2) - 20*diff(u);
L.bc = 'dirichlet'; LW = 'linewidth';
hold off, plot(u0,LW,1), grid on
```



Now we can use `expm` to compute $\exp(tL)$ for any time t. For example, here are the results at times 0.1 and 0.2:

```
hold on
u2 = expm(L,0.1,u0); plot(u2,'r',LW,1)
u4 = expm(L,0.2,u0); plot(u4,'m',LW,1)
```



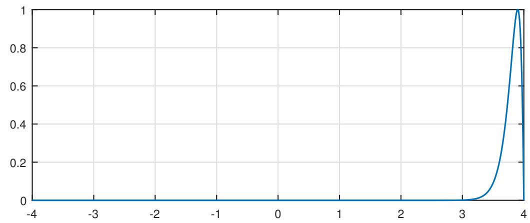
Here are the results at various times:

```
for t = 0.1:.1:.4
      u = expm(L,t,u0);
      [val,pos] = max(u); disp([t val pos])
end
```

```
    0.1000     0.6471     2.0000
    0.2000     0.4991     3.7966
    0.3000     0.0201     3.8757
    0.4000     0.0000     3.8882
```

This suggests that the answer is around 3.9. Alternatively the problem can be solved exactly with a little analysis. As $t \to \infty$ the solution approaches the dominant eigenfunction of the differential operator, which is

```
f = exp(10*x).*cos(pi*x/8); f = f/max(f);
hold off, plot(f,LW,1)
```



The maximum is easily found:

```
[val,pos] = max(f); pos

pos = 3.9000514
```

6. Heat equation on a square

We can use a code like this, adapted from `m48_CrankNicolson2D.m`, to get an idea of the solution:

```
J = 32; h = 1/J; s = (h:h:1)'; k = .0001;
[xx,yy] = meshgrid(s,s);
x = xx(:); y = yy(:);
u = double(abs(x-.5)<.25 & abs(y-.5)<.25);
I = speye(J); II = speye(J^2);
D = h^(-2)*toeplitz([-2 1 zeros(1,J-2)]);
L = kron(I,D) + kron(D,I);
A = II + k*L/2; B = II - k*L/2;
t = 0;
while max(u) > .5
      t = t+k; u = B\(A*u);
end
t

t = 0.024900
```

On taking $J = 4, 8, 16, 32, \ldots$ and halving the time step for extra confidence, we find apparent linear convergence to a critical time of $t = 0.0281$. The details are not shown here.

Actually this problem can perhaps better be solved by Fourier analysis. Any heat distribution can be reduced to a linear combination of eigenfunctions, of which the lowest component is $A\sin(\pi x)\sin(\pi y)$ with $A = 8/\pi^2$. This component will decay in time at the rate $C(t) = \exp(-2\pi^2 t)A$, and if this were the only component of the problem, we could solve $\exp(-2\pi^2 t)A = 0.5$ to find $t = \log(2A)/(2\pi^2) = \log(16/\pi^2)/(2\pi^2) = 0.0245$. Bringing in the next few terms of the series involving $\sin(2\pi x)\sin(\pi y)$, $\sin(\pi x)\sin(2\pi y)$ etc. would give quick convergence to the correct result.