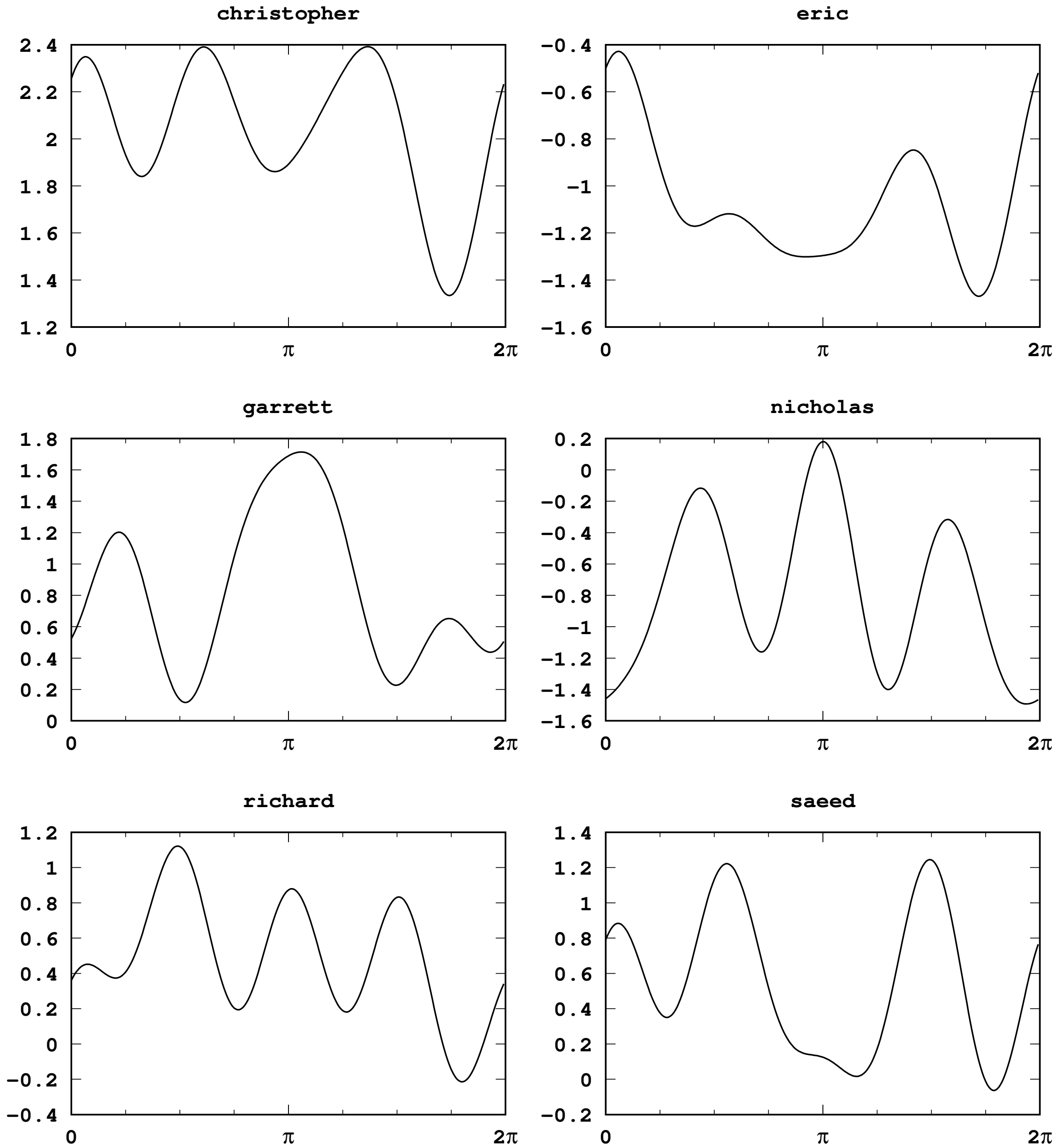
The Viscous Burgers Equation

1. Plot the points in the data file that correspond to your initial condition.



2. Energy analysis shows that the velocity profile given by the solution $u(t, x)$ to the viscous Burgers equation converges to a constant as $t \to \infty$. Find that constant for your initial condition.

Let the constant be denoted by V. Then V is equal to the average value of the initial condition which may be computed as

$$V = \frac{1}{L} \int_0^L u_0(x)dx \approx \frac{1}{N} \sum_{i=1}^{N} u_i$$

where the u_i are the data points in the data file that corresponds to the initial condition. MATLAB code to compute V looks like

```
load("eric.dat");
y=eric(:,1)';
sum(y)/length(y)
```

Running this script on each of the initial conditions yields

	Average
christopher.dat	2.0288
eric.dat	−1.07014
garrett.dat	0.836337
nicholas.dat	−0.77273
richard.dat	0.480252
saeed.dat	0.543693

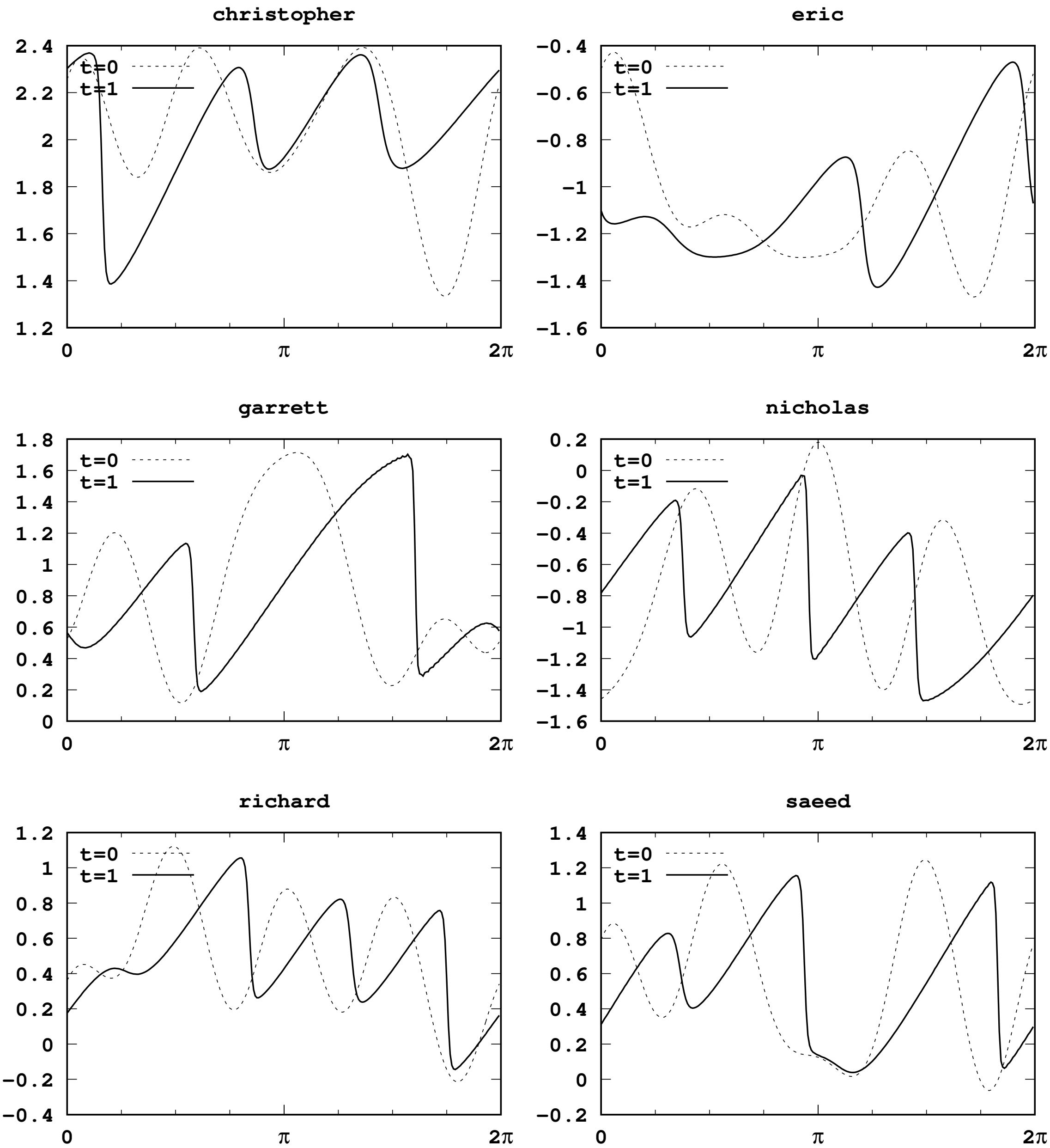
3. For Fourier transforms of length $N = 256$ show that the 2/3 anti-aliasing rule gives a cutoff with $K/2 = 85$.

The 2/3 anti-aliasing rule arrises by taking $K = \alpha N$ where $\alpha < 2/3$. Since $N = 256$ then

$$K/2 = \alpha N \leq (1/3)256 = 85.\overline{3}.$$

Since 85 is the greatest integer less than or equal $85.\overline{3}$ then we take $K/2 = 85$.

4. Use the RK2 method with a time step of $h = 1/128$ and $\varepsilon = 0.01$ to compute an approximation of $u(t, x)$ at time $t = 1$. Plot your approximation.



5. Given a time step of size $h = 2^{-m}$ let $U_m(x_\ell)$ be the corresponding approximation of $u(t,x)$ at time $t = 1$. Compute the norms of the errors

$$E_m = \|U_m - U_{m+1}\| \qquad \text{for} \qquad m = 6, \ldots, 11.$$

and make a table showing E_m versus m.

The Matlab script

```
clear all
global kvec filt epsilon
N=256;
K=2*floor(N/3);
kvec=[0:K/2,zeros(1,N-K-1),-K/2:-1];
filt=[ones(1,K/2+1),zeros(1,N-K-1),ones(1,K/2)];
epsilon=0.01;
Tfin=1;
x=[0:2*pi/N:2*pi-2*pi/N];
load('eric.dat');
y=eric(:,1)';
a0=ifft(y).*filt;
yold=y;
fprintf("eric\n%8s %12s\n","m","E_m");
for steps=2.^[6:11]
    h=Tfin/steps;
    at=a0;
    for j=1:steps
        k1=h*f(at);
        k2=h*f(at+k1);
        at=at+(k1+k2)/2;
    end
    yt=real(fft(at));
    Em=norm(yt-yold);
    fprintf(stdout,"%8d %12.4e\n",steps,Em);
    yold=yt;
end
```

produced the following outputs for each of the indicated initial conditions:

```
christopher
     2^m          E_m
      64          NaN
     128          NaN
     256   4.7537e-02
     512   1.2081e-02
    1024   3.0184e-03
    2048   7.5113e-04
```

```
eric
     2^m          E_m
      64   6.3288e+00
     128   1.4254e-02
     256   3.5270e-03
     512   8.7629e-04
    1024   2.1823e-04
    2048   5.4437e-05
```

garrett

2^m	E_m
64	NaN
128	NaN
256	1.8480e-02
512	4.5682e-03
1024	1.1141e-03
2048	2.7372e-04

nicholas

2^m	E_m
64	1.0847e+01
128	4.2760e-02
256	1.0800e-02
512	2.6357e-03
1024	6.4498e-04
2048	1.5917e-04

richard

2^m	E_m
64	6.1477e+00
128	1.0869e-02
256	2.6597e-03
512	6.5088e-04
1024	1.6057e-04
2048	3.9854e-05

saeed

2^m	E_m
64	8.1352e+00
128	2.1994e-02
256	5.3688e-03
512	1.2988e-03
1024	3.1788e-04
2048	7.8549e-05

6. Form the ratios E_m/E_{m+1} from the values found in the previous problem to verify that your implementation of the RK2 method is actually second order. Explain the reasoning behind your verification.

christopher

m	E_m	E_m/E_m+1
6	NaN	NaN
7	NaN	NaN
8	4.7537e-02	3.9350
9	1.2081e-02	4.0023
10	3.0184e-03	4.0184
11	7.5113e-04	

eric

m	E_m	E_m/E_m+1
6	6.3288e+00	443.9905
7	1.4254e-02	4.0415
8	3.5270e-03	4.0250
9	8.7629e-04	4.0155
10	2.1823e-04	4.0088
11	5.4437e-05	

garrett

m	E_m	E_m/E_m+1
6	NaN	NaN
7	NaN	NaN
8	1.8480e-02	4.0454
9	4.5682e-03	4.1002
10	1.1141e-03	4.0703
11	2.7372e-04	

nicholas

m	E_m	E_m/E_m+1
6	1.0847e+01	253.6708
7	4.2760e-02	3.9591
8	1.0800e-02	4.0977
9	2.6357e-03	4.0865
10	6.4498e-04	4.0521
11	1.5917e-04	

richard

m	E_m	E_m/E_m+1
6	6.1477e+00	565.6396
7	1.0869e-02	4.0864
8	2.6597e-03	4.0863
9	6.5088e-04	4.0535
10	1.6057e-04	4.0290
11	3.9854e-05	

saeed

m	E_m	E_m/E_m+1
6	8.1352e+00	369.8754
7	2.1994e-02	4.0967
8	5.3688e-03	4.1335
9	1.2988e-03	4.0860
10	3.1788e-04	4.0469
11	7.8549e-05	

If the method is second order we reason that

$$U_m - u(1,\cdot) \approx Mh^2 \qquad \text{where} \qquad h = 2^{-m}$$

for some vector $M \in \mathbf{R}^N$. It follows that

$$\begin{aligned} E_m &= \|U_m - U_{m+1}\| = \big\|\big(U_m - u(1,\cdot)\big) - \big(U_{m+1} - u(1,\cdot)\big)\big\| \\ &\approx \|Mh^2 - M(h/2)^2\| = (3/4)\|M\|h^2 = C4^{-m} \end{aligned}$$

where $C = (3/4)\|M\|$. Consequently,

$$\frac{E_m}{E_{m+1}} \approx \frac{C4^{-m}}{C4^{-m-1}} = \frac{1}{4^{-1}} = 4.$$

Now, since the calculated ratios E_m/E_{m+1} approach 4 in the output for each of the initial conditions considered above, we conclude that our implementation of the RK2 method is actually second order.