```matlab
%%%%%%%%%%%%%%%%%%%% Global Variables %%%%%%%%%%%%%%%%%%%%
global nodeco  elnode  bdynde  bdyedge  nVert  nedge
global GlobalV  GlobalP  GlobalS  GlobalG
global dimTvel  dimTpre  dimTstr  dimTGrv
global vel_bas_type  pre_bas_type  str_bas_type  Grv_bas_type
global quad_rul num mesh

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

vel_bas_type = 'CtsLin' ;
quad_rul = 'quad_73';
num = 1;
listmesh = ["R4x4","R6x6","R8x8","R10x10","R12x12"];

l = zeros(5,5);
for i=1:5
    mesh = listmesh(i);
    [uL2Error, GraduL2Error, H1uError]=Drv_ConDiff;
    l(i,:)=[uL2Error,2*i+2, 0,H1uError, 0];
end
for i=1:4
    l(i+1,3) = log(l(i,1)/l(i+1,1))/log(l(i+1,2)/l(i,2));
    l(i+1,5) = log(l(i,4)/l(i+1,4))/log(l(i+1,2)/m(i,2));
end
lin=table(l(:,2),l(:,1),l(:,3),l(:,4),l(:,5));
lin.Properties.VariableNames=["Mesh","uL2Error","alpha
uL2","H1uError","alpha H1u"];

disp("Part 2")
disp("Using Continuous Linears")
disp(lin)
disp("Thus the experimental convergence rate is about 1 for H_1 norm, which
is what we would expect since" +...
    " the error in the H_1 norm should be in the first term because we are
using continuous linear aproximations" + ...
    " and the error for the L2 norm is about 2 which is again what we would
expect using continuous linears.")
disp("I believe that my program works correctly as it solves the given
problems to the error" + ...
    " that we would expect given the continuous linear approximation.")

vel_bas_type = 'CtsQuad' ;
l = zeros(5,5);
for i=1:5
    mesh = listmesh(i);
    [uL2Error, GraduL2Error, H1uError]=Drv_ConDiff;
    l(i,:)=[uL2Error,2*i+2, 0,H1uError, 0];
end
for i=1:4
    l(i+1,3) = log(l(i,1)/l(i+1,1))/log(l(i+1,2)/l(i,2));
    l(i+1,5) = log(l(i,4)/l(i+1,4))/log(l(i+1,2)/m(i,2));
end
```

```matlab
quad=table(l(:,2),l(:,1),l(:,3),l(:,4),l(:,5));
quad.Properties.VariableNames=["Mesh","uL2Error","alpha
uL2","H1uError","alpha H1u"];


disp("Using Continuous Quadratics")
disp(quad)

disp("I believe that my program works correctly as it solves the given
problems to machine error" + ...
    " which is what we would expect given the continuous quadratic
approximation on a quadratic solution.")


vel_bas_type = 'CtsLin' ;
quad_rul = 'quad_73';
num = 0;

l = zeros(5,5);
for i=1:5
    mesh = listmesh(i);
    [uL2Error, GraduL2Error, H1uError]=Drv_ConDiff;
    l(i,:)=[uL2Error,2*i+2, 0,H1uError, 0];
end
for i=1:4
    l(i+1,3) = log(l(i,1)/l(i+1,1))/log(l(i+1,2)/l(i,2));
    l(i+1,5) = log(l(i,4)/l(i+1,4))/log(l(i+1,2)/m(i,2));
end
lin=table(l(:,2),l(:,1),l(:,3),l(:,4),l(:,5));
lin.Properties.VariableNames=["Mesh","uL2Error","alpha
uL2","H1uError","alpha H1u"];

disp("Part 3a")
disp("Using Continuous Linears")
disp(lin)
disp("Thus the experimental convergence rate is about 1 for H_1 norm, which
is what we would expect since" +...
    " the error in the H_1 norm should be in the first term because we are
using continuous linear aproximations" + ...
    " and the error for the L2 norm is about 2 which is again what we would
expect using continuous linears.")

vel_bas_type = 'CtsQuad' ;
l = zeros(5,5);
for i=1:5
    mesh = listmesh(i);
    [uL2Error, GraduL2Error, H1uError]=Drv_ConDiff;
    l(i,:)=[uL2Error,2*i+2, 0,H1uError, 0];
end
for i=1:4
    l(i+1,3) = log(l(i,1)/l(i+1,1))/log(l(i+1,2)/l(i,2));
    l(i+1,5) = log(l(i,4)/l(i+1,4))/log(l(i+1,2)/m(i,2));
end
quad=table(l(:,2),l(:,1),l(:,3),l(:,4),l(:,5));
```

```matlab
quad.Properties.VariableNames=["Mesh","uL2Error","alpha
uL2","H1uError","alpha H1u"];


disp("Using Continuous Quadratics")
disp(quad)

disp("Thus the experimental convergence rate is about 2 for H_1 norm, which
is what we would expect since" +...
    " the error in the H_1 norm should be in the second term because we are
using continuous quadratic aproximations" + ...
    " and the error for the L2 norm is about 3 which is again what we would
expect using continuous quadratics.")


quad_rul = 'quad_75';

l = zeros(5,5);
for i=1:5
    mesh = listmesh(i);
    [uL2Error, GraduL2Error, H1uError]=Drv_ConDiff;
    l(i,:)=[uL2Error,2*i+2, 0,H1uError, 0];
end
for i=1:4
    l(i+1,3) = log(l(i,1)/l(i+1,1))/log(l(i+1,2)/l(i,2));
    l(i+1,5) = log(l(i,4)/l(i+1,4))/log(l(i+1,2)/m(i,2));
end
quad=table(l(:,2),l(:,1),l(:,3),l(:,4),l(:,5));
quad.Properties.VariableNames=["Mesh","uL2Error","alpha
uL2","H1uError","alpha H1u"];

disp("Part 3b")
disp("Using Continuous Quadratics")
disp(quad)

disp("I did not see a difference in the experimental convergence rate after
changing the quad rule." + ...
    " If anything, it got worse.")
```

Part 2
Using Continuous Linears

Mesh	uL2Error	alpha uL2	H1uError	alpha H1u
4	0.1256	0	1.2966	0
6	0.056026	1.991	0.86088	1.0101
8	0.031547	1.9964	0.64457	1.0059
10	0.020196	1.9987	0.5152	1.004
12	0.014025	1.9999	0.4291	1.003

*Thus the experimental convergence rate is about 1 for H_1 norm, which is
what we would expect since the error in the H_1 norm should be in the first
term because we are using continuous linear aproximations and the error for
the L2 norm is about 2 which is again what we would expect using continuous*

linears.

I believe that my program works correctly as it solves the given problems to the error that we would expect given the continuous linear approximation.

Using Continuous Quadratics

Mesh	uL2Error	alpha uL2	H1uError	alpha H1u
4	3.047e-15	0	2.5158e-14	0
6	1.3038e-14	-3.5854	1.1881e-13	-3.8286
8	8.864e-15	1.3414	7.6957e-14	1.5096
10	7.0471e-14	-9.2909	2.8889e-13	-5.9281
12	1.893e-14	7.2094	1.8826e-13	2.3487

I believe that my program works correctly as it solves the given problems to machine error which is what we would expect given the continuous quadratic approximation on a quadratic solution.

Part 3a

Using Continuous Linears

Mesh	uL2Error	alpha uL2	H1uError	alpha H1u
4	0.32222	0	3.2795	0
6	0.17828	1.4598	2.3855	0.78498
8	0.098058	2.0779	1.7869	1.0043
10	0.060215	2.1853	1.4122	1.0545
12	0.041956	1.9816	1.1701	1.0317

Thus the experimental convergence rate is about 1 for H_1 norm, which is what we would expect since the error in the H_1 norm should be in the first term because we are using continuous linear aproximations and the error for the L2 norm is about 2 which is again what we would expect using continuous linears.

Using Continuous Quadratics

Mesh	uL2Error	alpha uL2	H1uError	alpha H1u
4	0.092447	0	1.4015	0
6	0.019723	3.8101	0.51914	2.4493
8	0.0084326	2.9535	0.30287	1.8732
10	0.0046116	2.7047	0.205	1.7491
12	0.0028562	2.6276	0.14932	1.738

Thus the experimental convergence rate is about 2 for H_1 norm, which is what we would expect since the error in the H_1 norm should be in the second term because we are using continuous quadratic aproximations and the error for the L2 norm is about 3 which is again what we would expect using continuous quadratics.

Part 3b

Using Continuous Quadratics

Mesh	uL2Error	alpha uL2	H1uError	alpha H1u
4	0.08336	0	1.3742	0
6	0.018036	3.7755	0.50222	2.4825

```
8       0.008108     2.7791      0.29605      1.8371
10      0.004465     2.6735      0.20179      1.7178
12      0.0028011    2.5574      0.14771       1.711
```

I did not see a difference in the experimental convergence rate after changing the quad rule. If anything, it got worse.

Published with MATLAB® R2023b