These days I tried soft clip.

So for the Beltrami regularization, the energy function becomes:

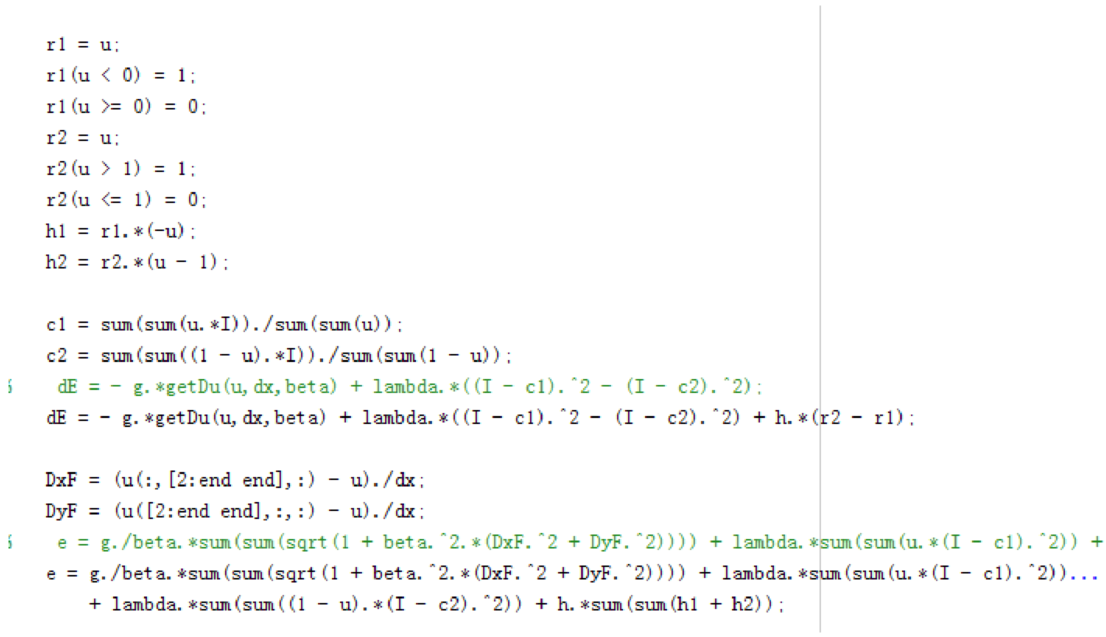
$$\mathrm{Ec(u)} = \int_\Omega \frac{1}{\beta}\sqrt{1+\|\beta u\|^2} + \lambda\int_\Omega u(I-c_1)^2 + \lambda\int_\Omega (1-u)(I-c_2)^2 + h\int_\Omega (h1+h2)$$

The gradient flow is:

$$\nabla E = -\mu\nabla\cdot\left(\frac{\beta\nabla u}{\sqrt{1+\|\beta u\|^2}}\right) + \lambda[(I-c1)^2-(I-c2)^2] + h(r2-r1)$$

Where h is the soft clip weight, r2 is the penalization for level set above 1, r1 is the penalization for level set below 0.

For ramp penalization, it doesn't work no matter how I choose h. If the h is too large, the level set cannot converge. If the h is too small, it cannot hold level set between 0 and 1. The code for ramp penalization is shown below.



```
    r1 = u;
    r1(u < 0) = 1;
    r1(u >= 0) = 0;
    r2 = u;
    r2(u > 1) = 1;
    r2(u <= 1) = 0;
    h1 = r1.*(-u);
    h2 = r2.*(u - 1);

    c1 = sum(sum(u.*I))./sum(sum(u));
    c2 = sum(sum((1 - u).*I))./sum(sum(1 - u));
%     dE = - g.*getDu(u,dx,beta) + lambda.*((I - c1).^2 - (I - c2).^2);
    dE = - g.*getDu(u,dx,beta) + lambda.*((I - c1).^2 - (I - c2).^2) + h.*(r2 - r1);

    DxF = (u(:,[2:end end],:) - u)./dx;
    DyF = (u([2:end end],:,:) - u)./dx;
%     e = g./beta.*sum(sum(sqrt(1 + beta.^2.*(DxF.^2 + DyF.^2)))) + lambda.*sum(sum(u.*(I - c1).^2)) +
    e = g./beta.*sum(sum(sqrt(1 + beta.^2.*(DxF.^2 + DyF.^2)))) + lambda.*sum(sum(u.*(I - c1).^2))...
        + lambda.*sum(sum((1 - u).*(I - c2).^2)) + h.*sum(sum(h1 + h2));
```

For quadratic penalization, I think the optimal damping coefficient is:

$$a = \frac{2\pi}{n}\sqrt{\beta g + h}$$

where g is weight for regularization term, β is the coefficient in Beltrami regularization, n is the scale of the image $R^{n\mathrm{x}n}$. The maximum gradient amplifier is:

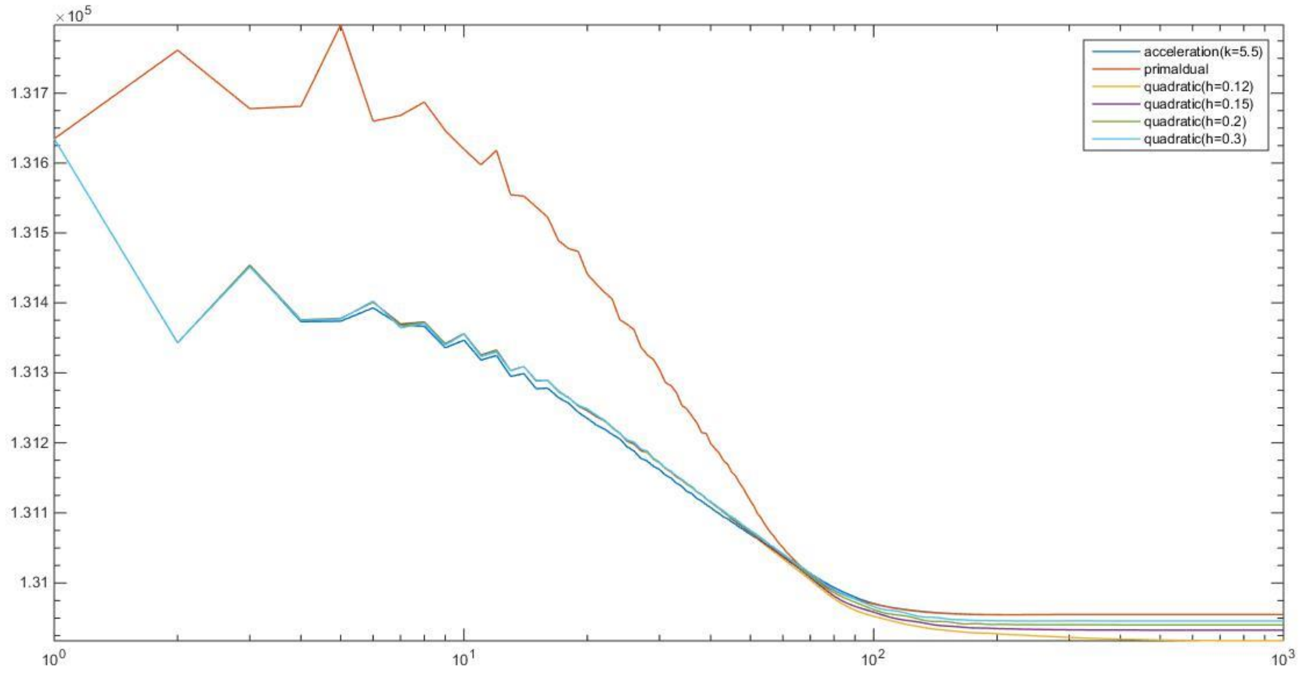
$$z_{\max} = \frac{4N\beta g}{\Delta x^2} + h$$

The code is shown below:

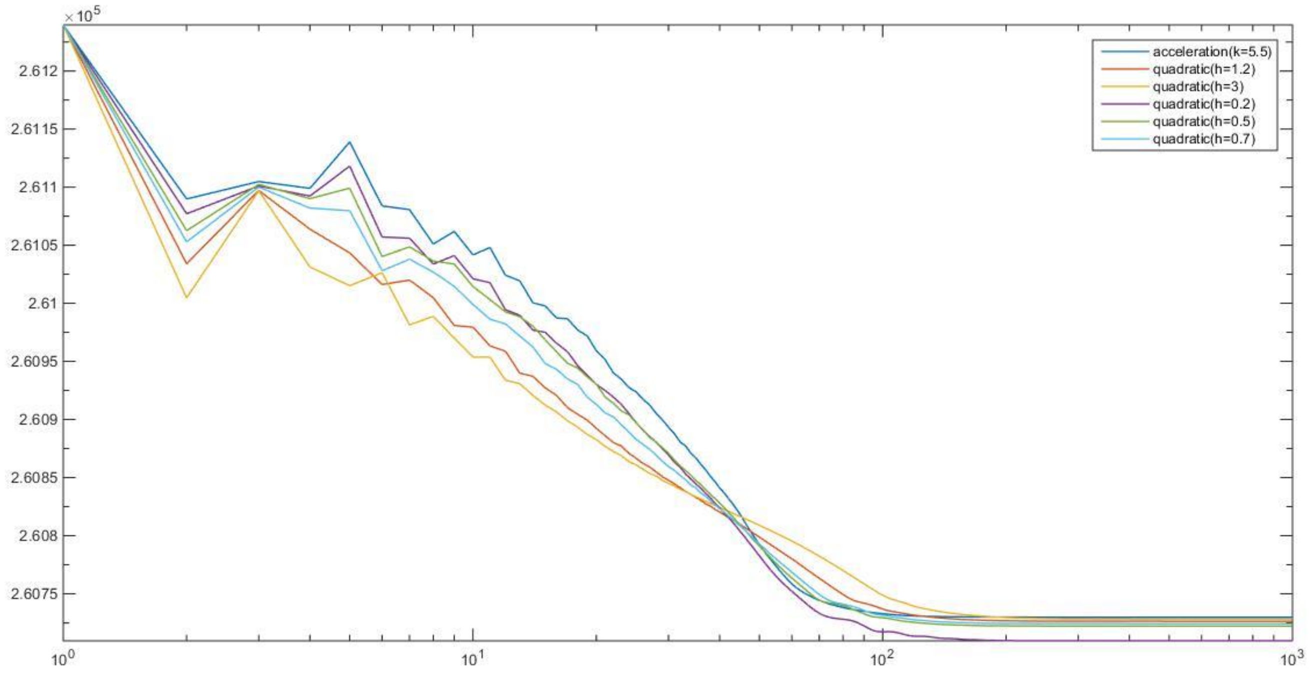
```
r1 = -u;
r1(r1 < 0) = 0;
r2 = u - 1;
r2(r2 < 0) = 0;
h1 = 0.5.*r1.^2;
h2 = 0.5.*r2.^2;


c1 = sum(sum(u.*I))./sum(sum(u));
c2 = sum(sum((1 - u).*I))./sum(sum(1 - u));
 dE = - g.*getDu(u,dx,beta) + lambda.*((I - c1).^2 - (I - c2).^2);
dE = - g.*getDu(u,dx,beta) + lambda.*((I - c1).^2 - (I - c2).^2) + h.*(r2 - r1);

DxF = (u(:,[2:end end],:) - u)./dx;
DyF = (u([2:end end],:,:) - u)./dx;
 e = g./beta.*sum(sum(sqrt(1 + beta.^2.*(DxF.^2 + DyF.^2)))) + lambda.*sum(sum(u.*(I - c1).^2)) + lambda.
e = g./beta.*sum(sum(sqrt(1 + beta.^2.*(DxF.^2 + DyF.^2)))) + lambda.*sum(sum(u.*(I - c1).^2))...
    + lambda.*sum(sum((1 - u).*(I - c2).^2)) + h.*sum(sum(h1 + h2));
```

The quadratic penalization did work. In the following experiment, I do not include soft penalization energy in the energy plot because it is more obvious to show how soft penalization is helpful for our original energy function. I did experiments for different β. And the damping coefficients used in the experiments are not optimal damping coefficients. I find the best damping coefficients by hand.

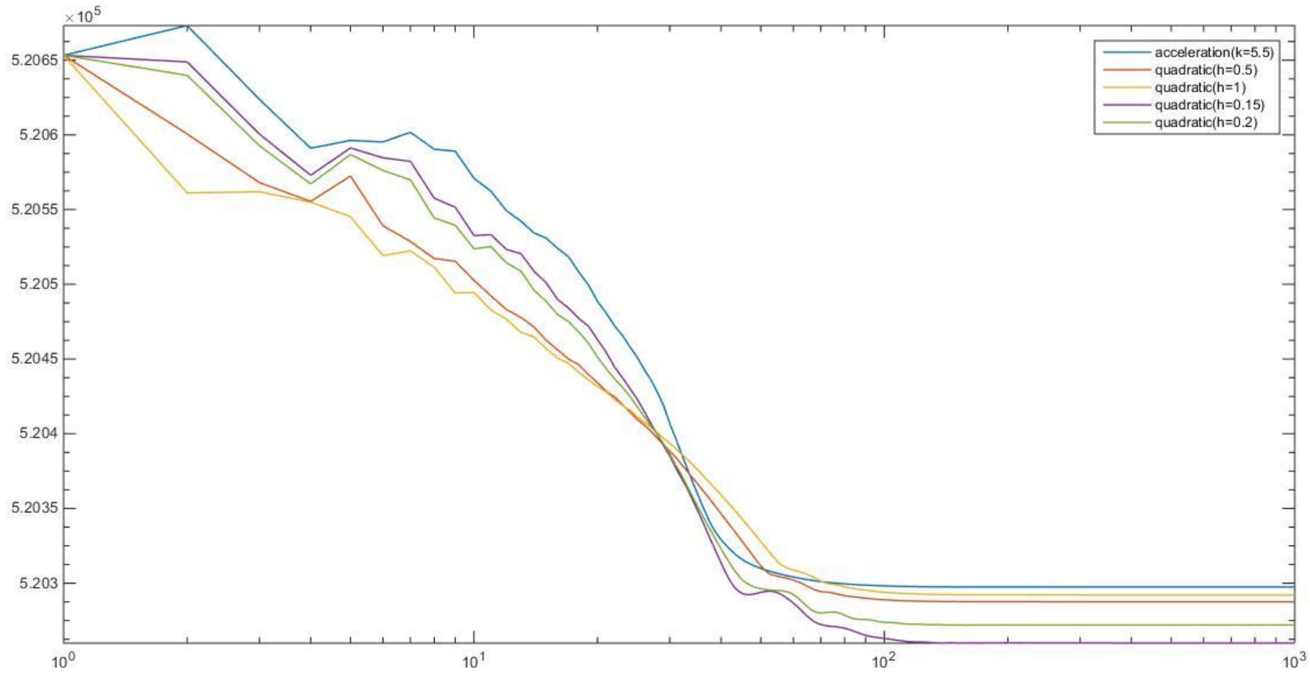
For β=2,



For β=1



For β=0.5

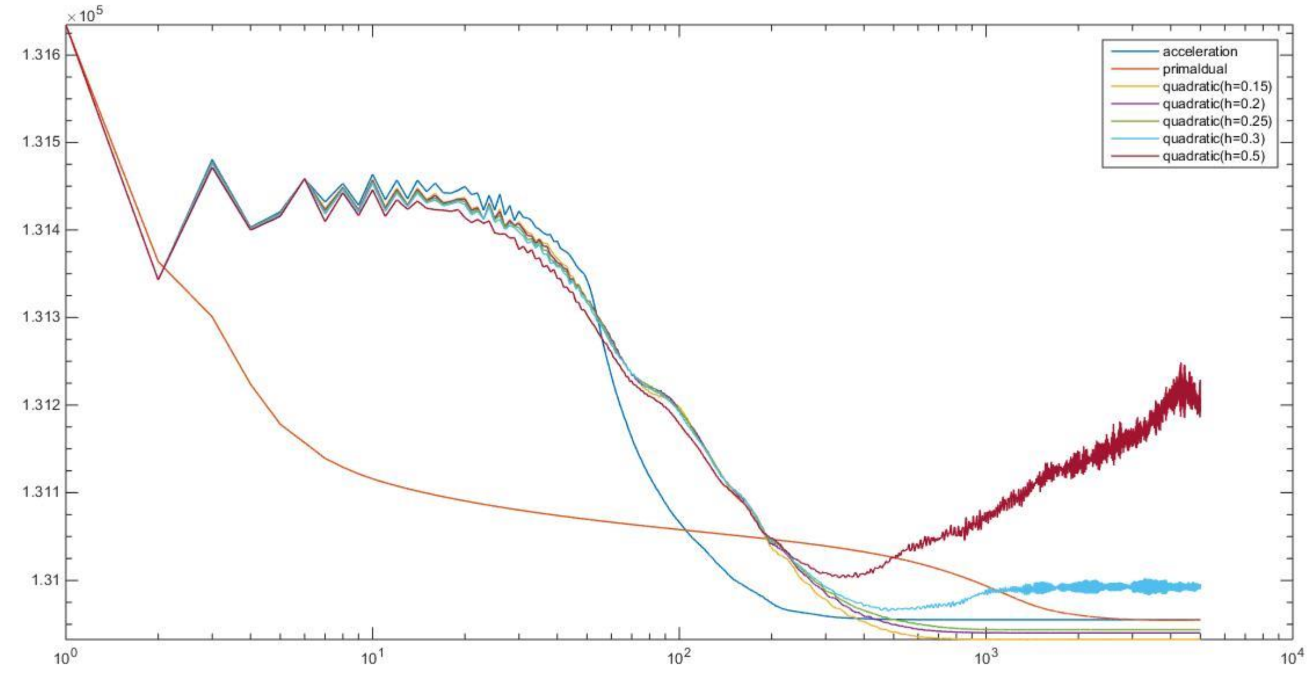


As we can see from the plots above, the smaller the soft penalization coefficient, the lower the final energy, but the convergence speed is slower.

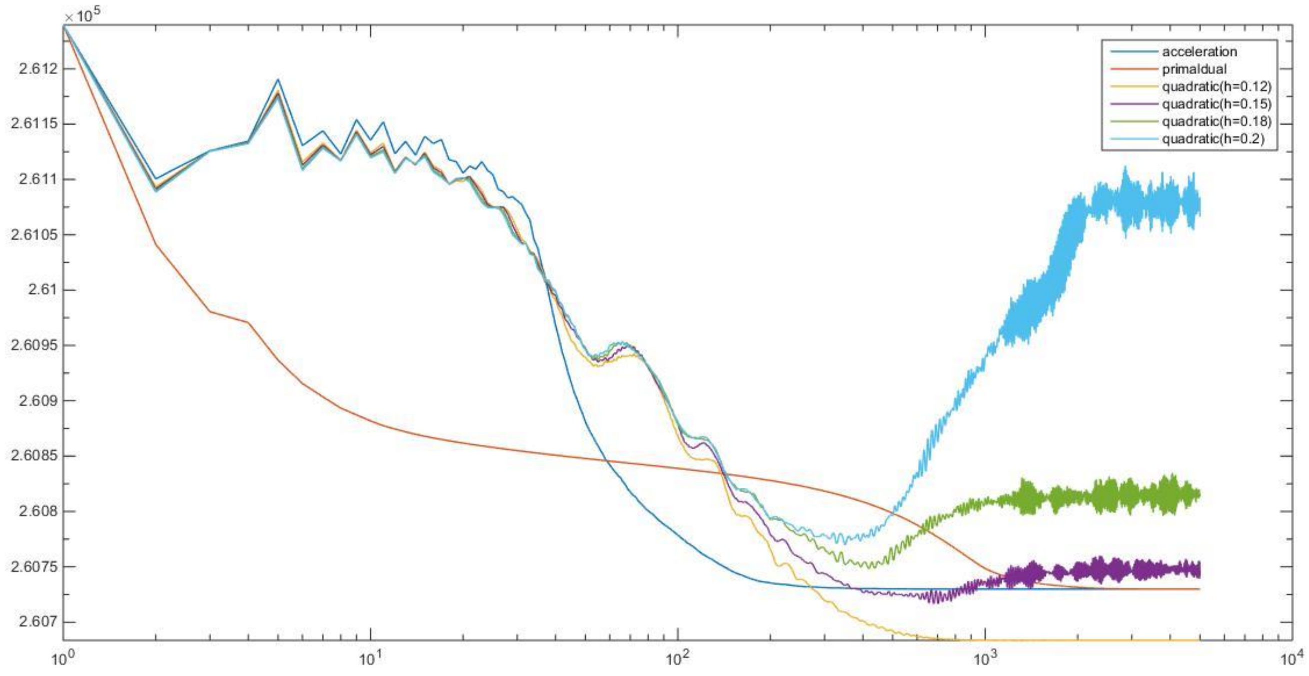
Now I think about that maybe I should not try to find the best damping coefficient by hand. Instead, I think we should use the theoretical optimal damping coefficient because in practice we cannot spend time find the best coefficient by hand each time. It is the same for Chambolle-Pock. There is a set of theoretical optimal parameters for Chambolle-Pock, but in the previous experiments I tried to find the best set of parameters by hand. I don't know if you use the theoretical optimal values all the time in your previous projects.

Now I start to do experiment with the theoretical optimal damping coefficient for acceleration and theoretical optimal parameters for Chambolle-Pock.

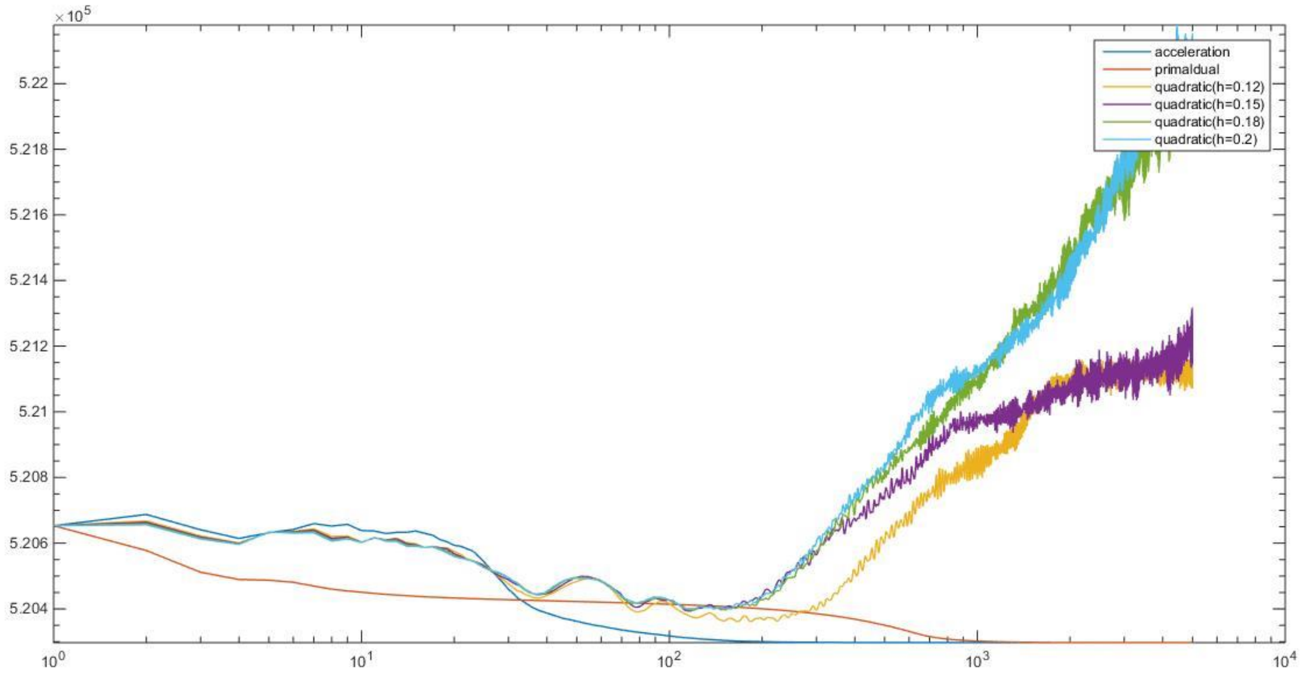
For β=2



For β=1



For β=0.5



If the damping coefficient is the theoretical optical damping coefficient, when the soft penalization weight is too large, the final energy will blow off. But as I shown in the last part, I find the best practical damping coefficient by hand, and with the practical best damping coefficient the final energy for soft penalization doesn't blow up. The practical best damping coefficient is much larger than theoretical optimal damping coefficient, so I start to think maybe I get the wrong theoretical optimal damping coefficient for soft penalization.
In summer you gave me the optimal damping coefficient for hard clip:

$$\mathrm{a} = \frac{2\pi}{n}\sqrt{\beta g}$$

So I get the optimal damping coefficient for soft clip:

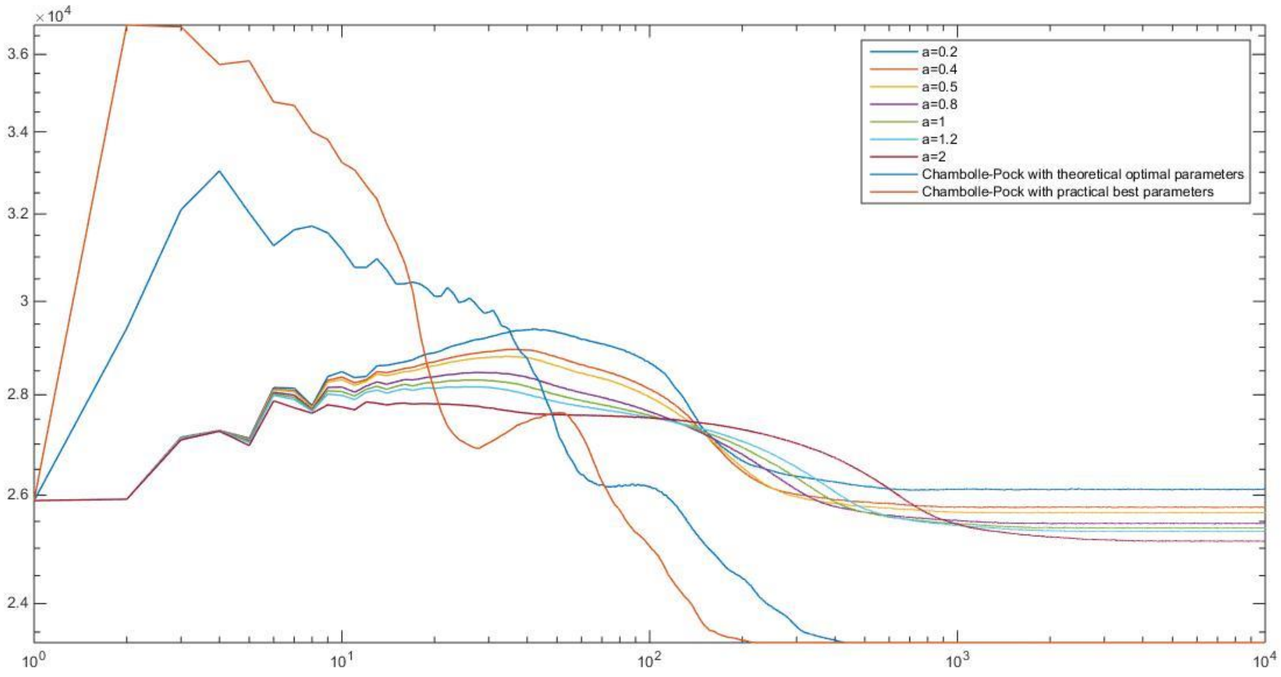
$$\mathrm{a} = \frac{2\pi}{n}\sqrt{\beta g + h}$$

Reminder the energy function for soft clip is:

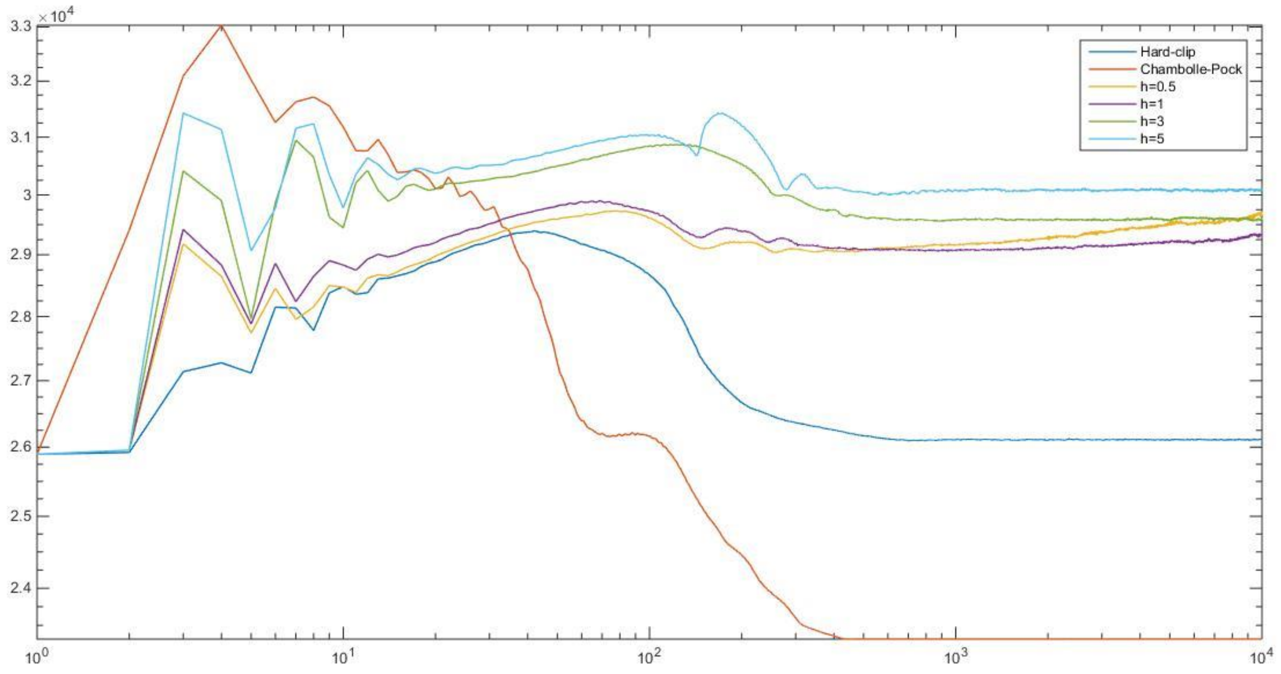
$$\mathrm{Ec(u)} = \int_{\Omega} \frac{1}{\beta}\sqrt{1 + \|\beta \mathrm{u}\|^2} + \lambda \int_{\Omega} \mathrm{u}(\mathrm{I} - \mathrm{c}_1)^2 + \lambda \int_{\Omega} (1 - \mathrm{u})(\mathrm{I} - \mathrm{c}_2)^2 + h\int_{\Omega} (h1 + h2)$$

Maybe you can help me check if my optimal damping coefficient for soft clip correct? Maybe the correct one is larger than it?

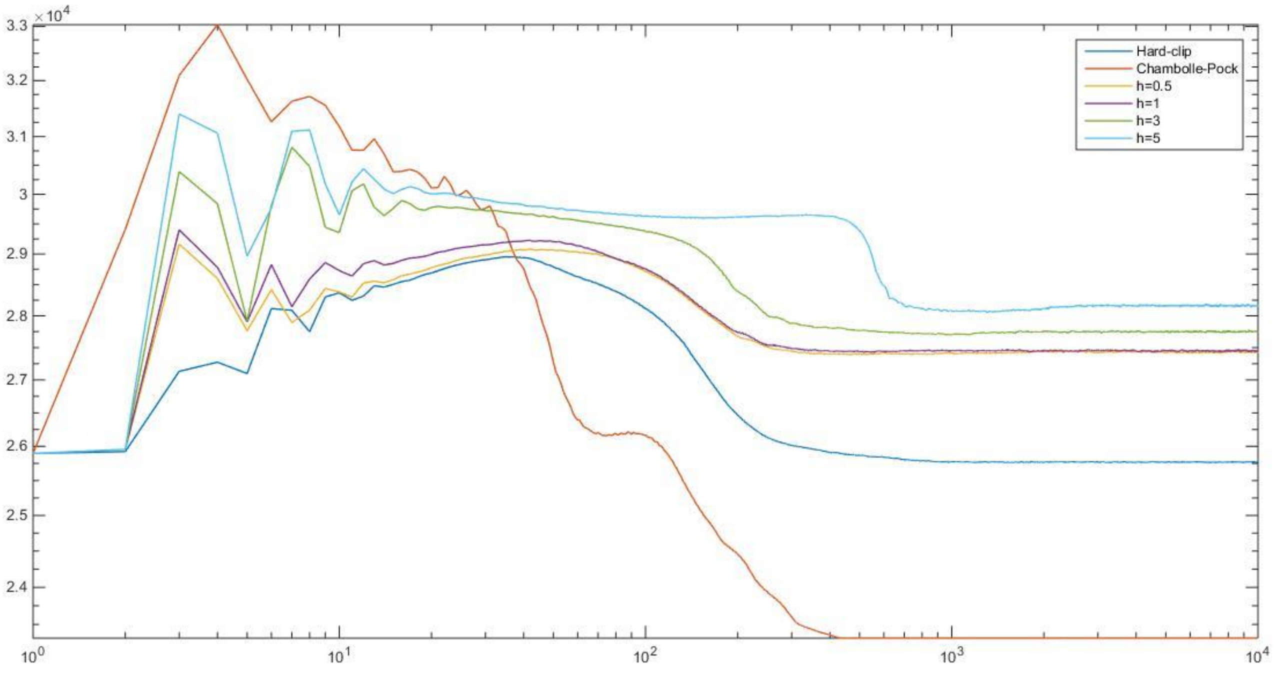
Now for TV regularization, an interesting thing is that in this case quadratic penalization doesn't work but ramp penalization works.
First I plot the energy for hard-clip and compare against Chambolle-Pock with theoretical optimal parameters and Chambolle-Pock with practical best parameters. a is the damping coefficient.



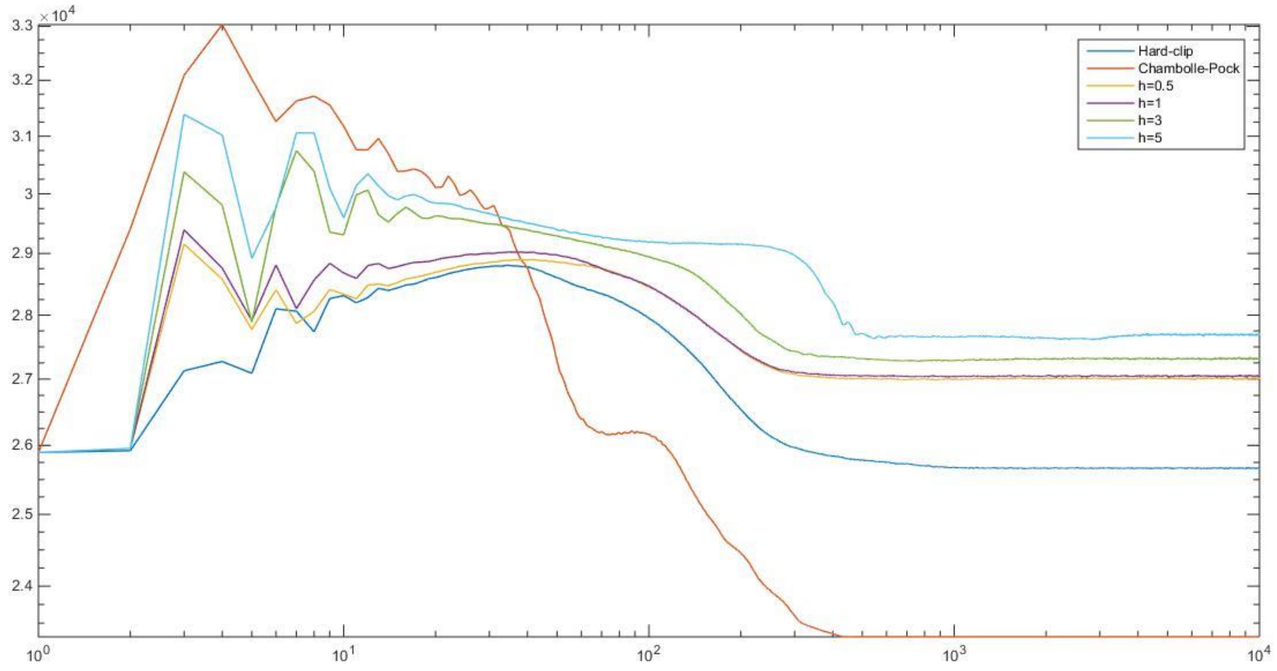
Now I start to do ramp penalization for TV regularization. I use quantization level Q=1/255.
For a =0.2



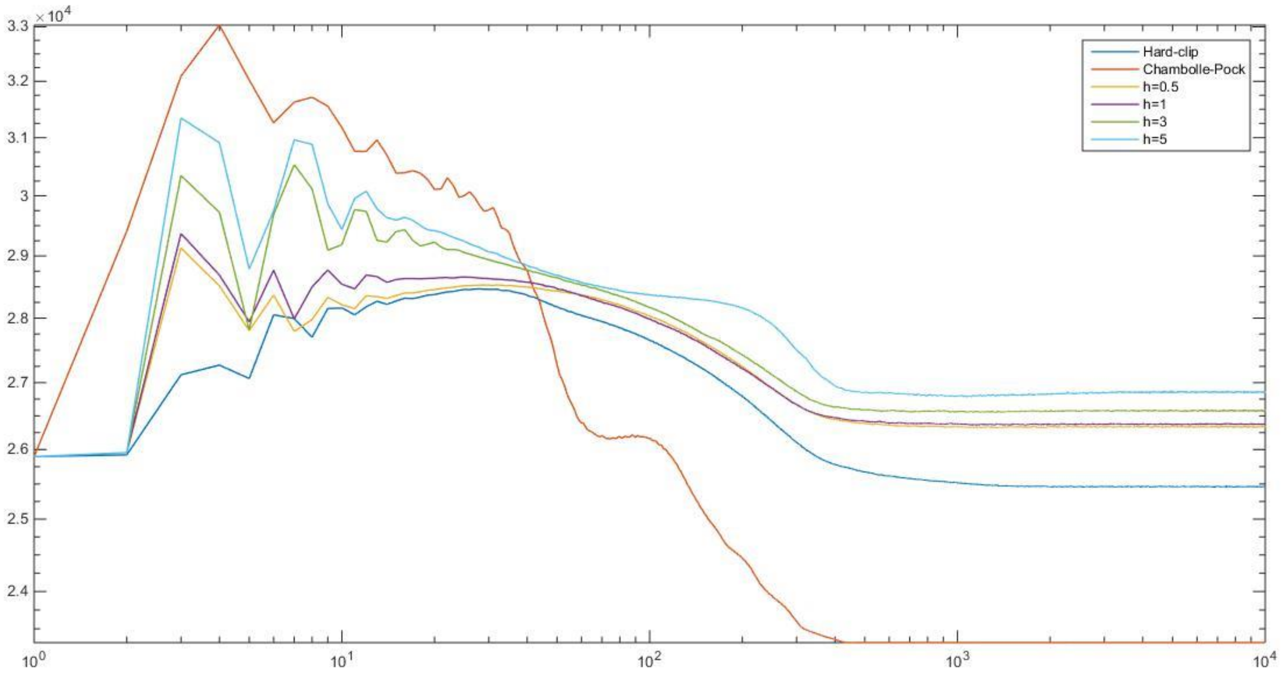
For a=0.4



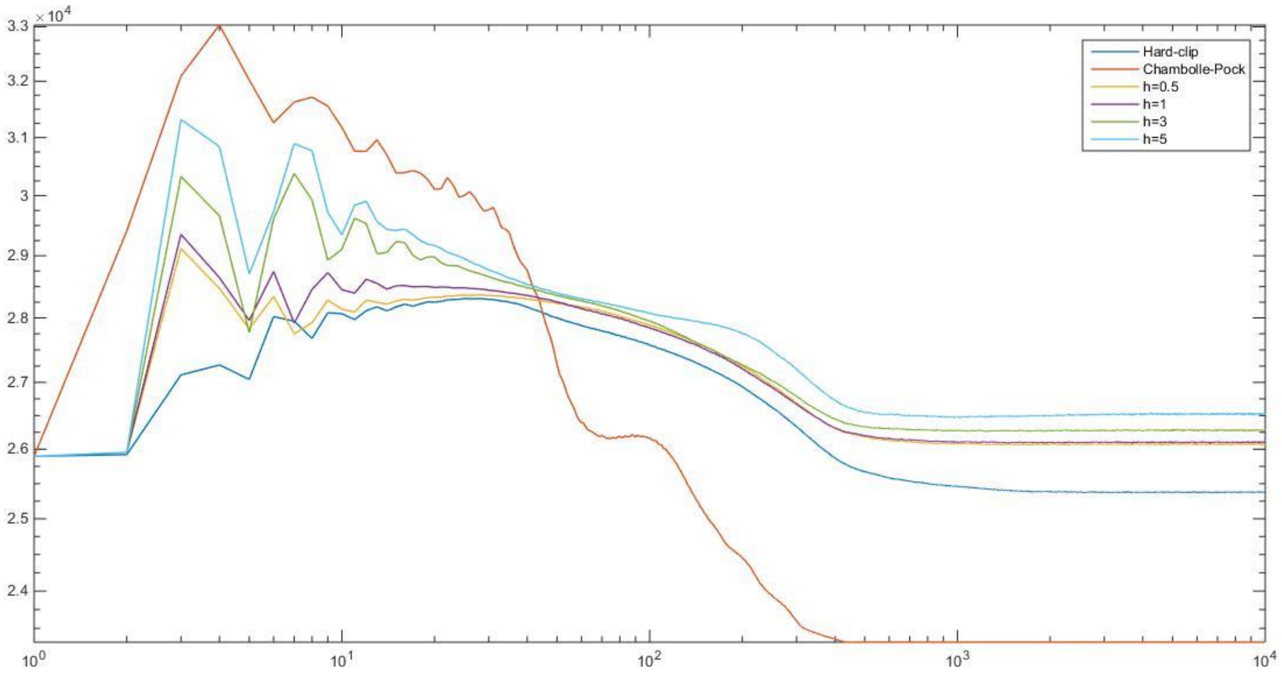
For a=0.5:



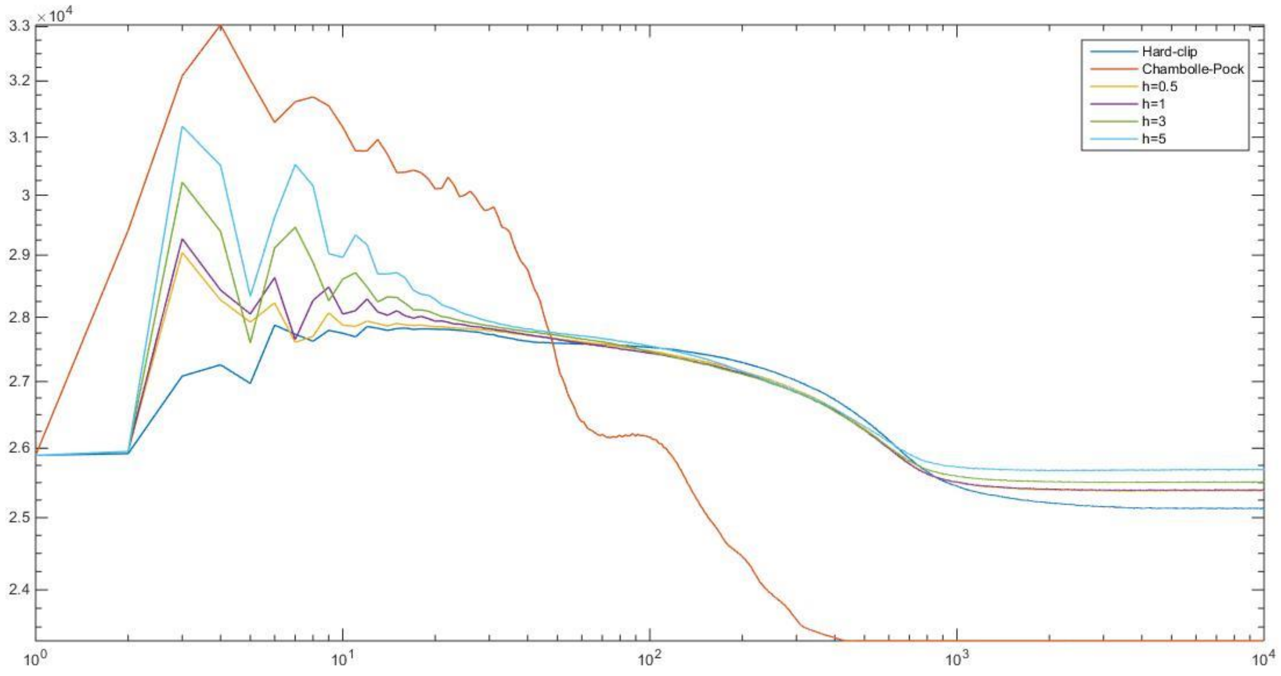
For a=0.8:



For a=1:



For a =2:



For TV regularization, it is normal that the final energy of acceleration for hard clip is larger than Chambolle-Pock because we set a quantization level Q in TV regularization and the algorithm is not exactly a PDE. The energy for soft clip is even larger because the surface will be more rugged without hard-clip and the final energy is violating much stronger. So there is no sense to compare the final energy in this case.

And all the plots done are energy against iteration plot rather than energy vs CPU time plot. I don't know how to plot energy against CPU time in matlab. However, I can measure the convergence time. So all of the energy plots shown in this report are just briefly indications of the performance of the algorithms.

In sum, for Beltrami regularization, if you think it is Ok to find the practical best damping coefficient for acceleration and the practical best set of parameter for Chambolle-Pock each time, then I continue to do experiment with the practical best values. Or if you think we should use the theoretical optimal values, then I continue with the practical optimal values. But I think I derive the wrong optimal damping coefficient for soft penalization; maybe you can help me get the right one?

For TV regularization, what's your opinion? In general I think Chambolle-Pock with the theoretical optimal set of parameters has a much better performance on TV regularization than Beltrami regularization. It converges faster and gets a better result on TV. Maybe it is because Chambolle-Pock algorithm is proposed for TV regularization in the original paper?