1 Numerical Results

1.1 Problem

diffusion equation:

$$\frac{\partial y}{\partial t} = \frac{\partial^2 y}{\partial x^2} - e^{-t}(\sin(\pi x) - \pi^2 \sin(\pi x)),\ x \in [1, -1], t \in [0, 1] \tag{1}$$

with the IC

$$y(x, 0) = \sin(\pi x)$$

and the Dirichlet BD

$$y(-1, t) = y(1, t) = 0$$

The solution is $y = e^{-t} \sin(\pi x)$

1.2 Numerical Results

Algorithm 1.1 Subsampled LM algorithm

Step 0 Random choose an parameter set B_0 and initial damping parameter $\mu_0 > 0$, constants $\gamma > 1$ and $k_{\max}$, parameter $p^0 \in R^n$, set $k = 0$, $l = 0$,

Step 1 if a stopping criteria is satisfied, stop; otherwise, go to Step 2.

Step 2 Solve the linear system and obtain the direction $h^k \in R^n$

Step 3 Compute the least square problem $F(p^k + h^k)$ and $F(p^k)$

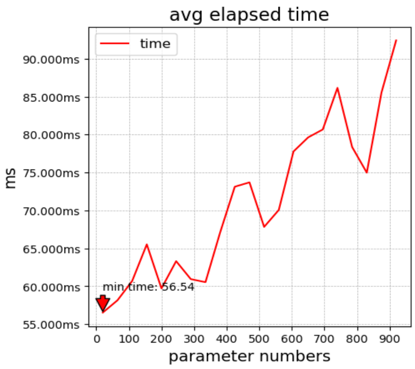
Step 4 if $F(p^k + h^k) < F(p^k)$, set $p^{k+1} = p^k + h^k$ and $\mu_{k+1} = \mu_k / \gamma$; otherwise set $p^{k+1} = p^k$ and $\mu^{k+1} = \gamma \mu^k$. Then $k = k + 1$

Step 5 if $k = k_{\max}$, choose a new parameter set B_l. Set k=0, $l = l + 1$, go back to Step 1. otherwise Go back to Step1.

To evaluate the performance of Subsample LM algorithm with different size of parameter sets $|B_l|$, I test it using a nerual network for solving diffusion equation(1). Recall the neural network $\varphi(x, t; p) = y$, the network widths are [2,20,20,20,1], 920 parmeters in total and the activation function $\tanh$ is used. Briefly, at each iteration of the LM method, the direction h^k is obtained by solving a linear system

$$(J^T J + \mu^k I) h^k = -J^T f$$

where $\mu^k > 0$ is damping parameter, I is the identity matrix, $J = \left(\frac{\partial f_i}{\partial p_j} \right)$ and f is the residue term. The algorithms with different subsample set sizes run with the same iterations(2000 iterations, resample data at every 100 iterations) and the same sample data. The average elapsed times are recorded in Figure 1 and the relative L2 error are recorded in Table 3. General speaking, small subsample size costs less time than the bigger size iteration.



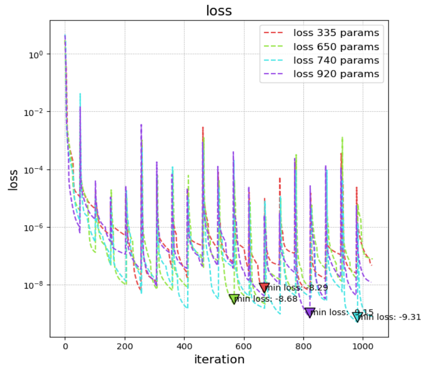


Figure 1(left): average elapsed time in terms of subsample size; Figure 2(right): loss of one run algorithm using 4 different sizes: 335, 650, 740, 920.

At the same time, the iteration with rational size has same or even better approximation accuracy comparable to that of tradition LM method. Figure 2 depicts the plots of the loss values for one run of algorithm with 4 sizes: 335, 650, 740, 920, all of which have similar L2 errors. Contrary to expectations, the larger size of the subsampled LM algorithm doesn't get a smaller losses or ralative L2 errors under same iterations. These results also illustrate that choosing a rational size for the parameter space of the LM algorithm may accelerate the algorithm's speed.

d	subsample size	relative L2 errors ($\times 10^{-2}$)
1	20	231.94
2	65	5.87
3	110	6.58
4	155	3.02
5	200	5.01
6	245	2.50
7	290	2.24
8	335	0.86
9	380	1.66
10	425	4.31
11	470	3.65
12	515	1.77
13	560	0.80
14	605	0.84
15	650	0.71
16	695	0.77
17	740	0.45
18	785	1.73
19	830	0.93
20	875	1.23
21	920	0.93

Table 3: Relative L2 errors

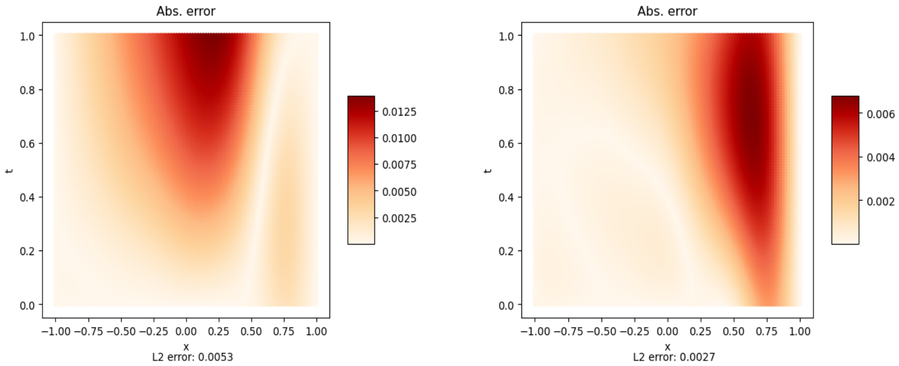


Figure 4. left: Scatter plot of the absolute error of the solution with 950 sizes. right: Scatter plot of the absolute error of the solution with 650 sizes . The L2 error is caculated by the 10000 uniform test points across $x \in [-1, 1], t \in [0, 1]$.

After repeating the experiment three times with two different subsample sizes, it was observed that the result with a subsample size of 650 consistently achieved a better L2 error than the traditional LM iterations. Two example outputs are plotted in Figure 4.