Optimization Algorithms Final Exam

Ayan Gangopadhyay

July 21, 2020

Problem Statement

We have been asked to minimize the Extended Rosenbrock function,

$$min \quad f(\mathbf{x}) = 100 \times \sum_{i=1}^{n-1} \left[(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right]$$

where $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ and $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

Using Conjugate Gradient methods we assume the starting point $\mathbf{x_0} = \{-1.2, 1, ..., -1.2, 1\}$ and $3 \leq n < 13$ and check the convergence for the *Fletcher-Reeves* (FR) and *Polak-Ribière-Poylak* (PRP) variants. We try to investigate if one has any advantage over the other in terms of convergence rate and accuracy as n increases.

Methods Used

The Conjugate Gradient method has been implemented using *Pytorch* which has been found to be superior to *numpy* for scientific computing escpecially in presence of a Graphics Processing Unit. The code was developed in part using the slides [1] and [2] provided by our instructor *Mrinmay Maharaj* and the book *Numerical Opimization by Nocedal & Wright [3].*

Seeing the results from various papers such as [4] and [5] it was seen that in general PRP methods work better than FR in case of non-linear optimization.

It was also indicated in [6], [7] and [8] that in some cases *preconditioning* the problem might help in convergence. But preconditioning was not implemented, and the reasoning is given later in the Appendix.

Theory

The non-linear conjugate gradient method was introduced by Fletcher and Reeves in [9]. They adapted the Conjugate Gradient method for minimization of functions instead of solving system of linear equations. Two simple changes were made to achieve this,

1. To identify the step length α we need to perform a line search that identifies an approximate minimum of the nonlinear function f along the descent direction p_k.

2. The residual is to be replaced by the gradient of the non-linear objective function f.

Algorithm

Since the objective function

$$min \quad f(\mathbf{x}) = 100 \times \sum_{i=1}^{n-1} \left[(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right]$$

is non-linear we use the algorithm proposed by Fletcher-Reeves to perform the minimization the algorithm can be specified as follows (taken from page 121 of [3]),

Non-Linear CG method

```
Given $x_0$;
Evaluate $f_0 = f(x_0), \nabla f_0 = \nabla f(x_0)$;
Set $p_0 \leftarrow -\nabla f_0$, $k \leftarrow 0$;
while $\nabla f_k \neq 0$
    Compute $\alpha_k$ and set $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
    Evaluate $\nabla f_{k+1}$;
    $\beta_{k+1} \leftarrow$ update $\beta$ (PRP or FR) ;
    $p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1} p_k$;
    $k \leftarrow k + 1$
```

Stopping Criteria

In the algorithm specified in the book the algorithm is terminated when $\nabla f_k = 0$, in practice this criterion was not found to be very useful. We put an upper limit to the maximum number of iterations and the algorithm was stopped when the distance between $\mathbf{x_k}$ and $\mathbf{x}^*$ was less than $2e^{-2}$ where $\mathbf{x_k}$ is the result of the k^{th} iteration and $\mathbf{x}^*$ is the global minima {1, 1, ..., 1}.

Implementation

For the Fletcher Reeves method it is recommended in [3] that a *restart-strategy* be implemented in order to achieve faster convergence. For the purpose of the demonstration we have shown the results for both FR method using a restart window of 10 iterations as well as FR method without a restart strategy.

Ideally any step length α that is taken should satisfy the following *strong Wolfe conditions,*

$$f(\mathbf{x_k} + \alpha_k \mathbf{p_k}) \leq f(\mathbf{x_k}) + c_1 \alpha_k \nabla f_k^T \mathbf{p_k}$$

and

$$|\nabla f(\mathbf{x_k} + \alpha_k \mathbf{p_k})| \leq -c_2 \nabla f_k^T \mathbf{p_k}$$

where $0 < c_1 < c_2 < \frac{1}{2}$. But there was no computationally efficient way to identify such a step length. It should be noted that, should such a step length be used in every iteration, the convergence of the Conjugate Gradient method (for any β update) is guaranteed. Instead to find the step length a backtracking line search method was used which produced slower convergence that ideal.

Results

Convergence Rate

To analyze the convergence rate the number of iterations and runtimes were recorded for each value of n. It was seen in general that the FR method had a higher runtime and needed more iterations to converge than the PRP method. This was in accordance to the results mentioned in [3] and [4].

The following plots were obtained when runtimes and the number of iterations required to converge were plotted as a function of the dimension of the problem n.

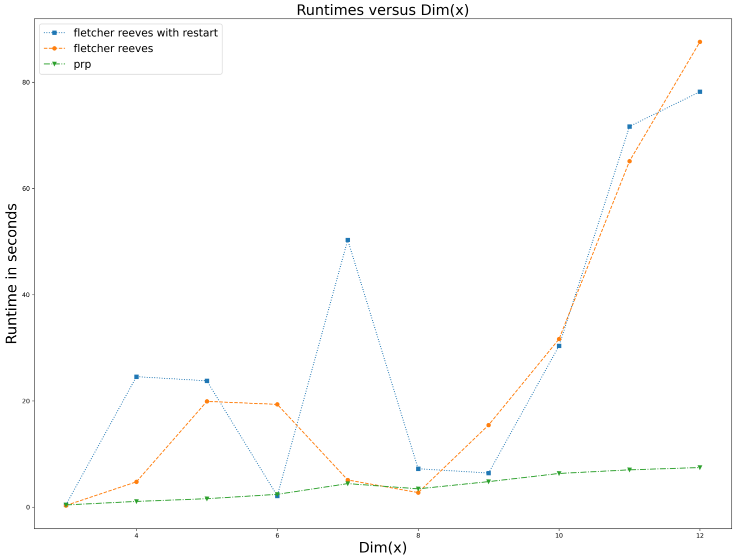

Figure 1:

Here we can see that the PRP method consistently gives runtimes which are less than those reported by the FR method.

The number of iterations needed to converge show a similar trend which can be seen in the following plot,

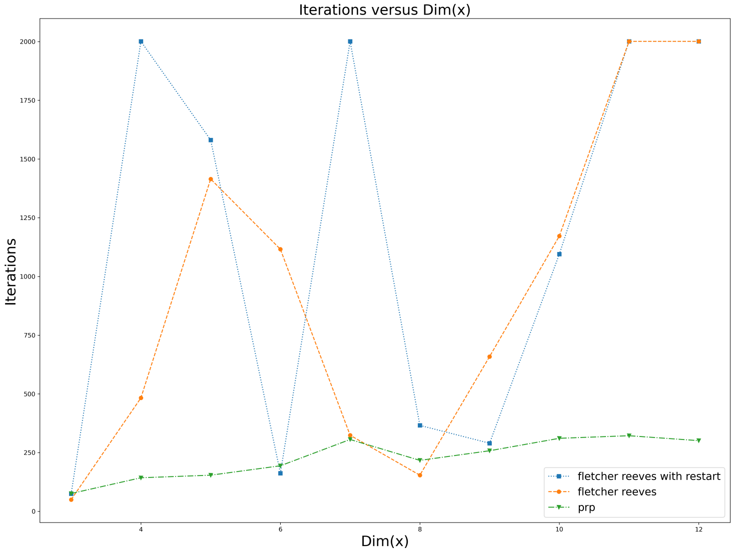

Figure 2:

From these two plots it can be seen that the PRP method outperforms both variants of the FR method. This is in accordance to the results in the papers mentioned earlier. We can see that as the value of n increases FR method becomes increasingly untenable in terms of convergence rate.

Accuracy

We know that for the Extended Rosenbrock function the true minima is $\mathbf{x}^*$ is $\{1, 1, ..., 1\}$. This can be seen easily from the form of the function where on plugging in all ones the function returns a value of 0 and otherwise it is positive.

To measure the accuracy of the results the distance from the true minima was plotted as a function of n, it was seen that for some values of n the distance shoots up in case of the FR method. The PRP method does not show such phenomena, although there are some instances of n for which both algorithms stop far away from the true minima. This can be attributed to the fact that the Hessian of the Rosenbrock function is quite ill-conditioned and convergence is very sensitive to starting points.

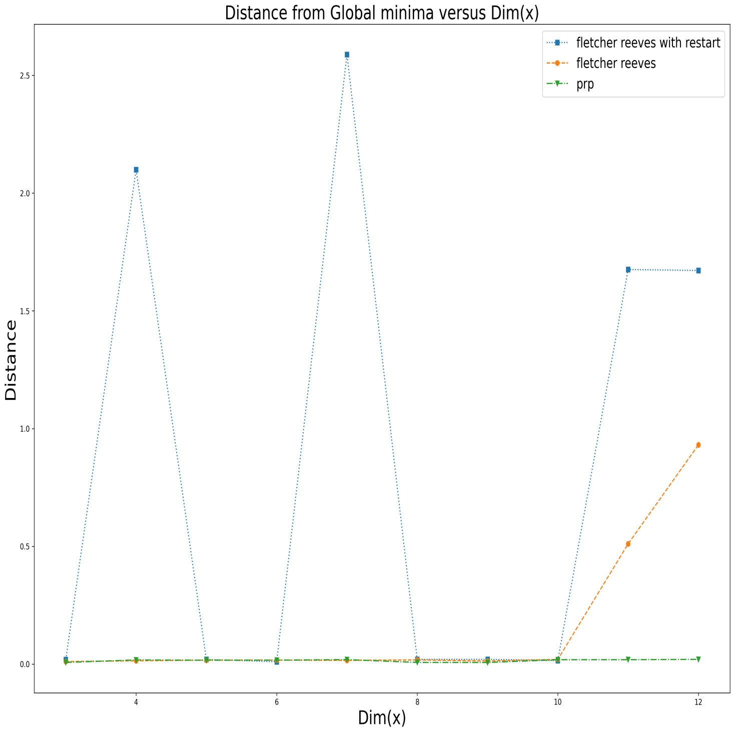

Figure 3:

Remarks

From the plots above we can see that the even if the restart strategy provides some improvement in convergence rate theoretically, when used with a *backtracking line search* it does not give an appreciable advantage over the normal FR method. Further, in terms of accuracy the FR method with restart strategy performs poorly even when compared to the vanilla version of FR method, whereas the PRP method handily beats both of these methods in terms of both accuracy and convergence rates.

From the above plots it is easy to see that the PRP method works better for the Extended Rosenbrock problem as n increases. This has been observed in general as well, that PRP method works well for most non-linear objective functions. But even the PRP method has its own drawbacks as Powell in 1984 [10] showed instances where the PRP method can cycle infinitely.

Whatever be the case, it seems like in light of all the evidence provided we can safely say that the PRP method is superior than FR method for the minimization of the Extended Rosenbrock function, and considering the high condition number for the Hessian for this curve, it is quite a welcome surprise to come across a method which performs well.

Appendix

Pre-Conditioning in Non-Linear Conjugate Gradient Method

A method to implement preconditioning in the non-linear version of the algorithm is given below and was decribed in [11].

Non-Linear CG method using Preconditioning

```
Given $x_0$;
Evaluate $f_0 = f(x_0), \nabla f_0 = \nabla f(x_0)$;
Solve $Py_0 = \nabla f_0$
Set $p_0 \leftarrow -y_0$, $k \leftarrow 0$;
while $\nabla f_k \neq 0$
    Compute $\alpha_k$ and set $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
    Evaluate $\nabla f_{k+1}$;
    $\beta_{k+1} \leftarrow$ update $\beta$ (PRP or FR) ;
    Solve $Py_{k+1} = \nabla f_{k+1}$
    $p_{k+1} \leftarrow -y_{k+1} + \beta_{k+1} p_k$;
    $k \leftarrow k + 1$
```

It requires either using the *exact* Hessian or an approximation of the Hessian, but since we are letting the values of n to be large, exact Hessian calculation becomes very ineffecient. Further, to apply preconditioning, we would need to solve the equation $Hy_k = \nabla f_k$ for each iteration of the Conjugate Gradient method (H is the Hessian matrix at point $\mathbf{x_k}$ and ∇f_k is the gradient of the function at the same point). This would be even more inefficient since the condition number of the Hessian matrix is high for the Extended Rosenbrock function.

The only way which would allow us to use preconditioning effectively be to use an approximation of the Hessian matrix B instead of the Hessian itself but more time will be required to find out exactly how to calculate this approximation efficiently and how to implement it. The various ways that were discussed in class during the quasi-Newton methods might provide a way to efficiently compute the Hessian approximation at each step but more time is needed to research and implement these methods for our problem.

The problem with doing all this, though, is that there is no clear indication that these methods lead to faster convergence when the value of n becomes very large, and if computing and using approximate Hessian representations will be very efficient at all in these cases. Only experimentation will give us insight as there is not much literature on this subject at the moment.

There is scope for improvement in this area and can be taken up as future work.

Bibliography

[1] Slide 1 on Conjugate Gradient Methods: Mrinmay Maharaj

[2] Slide 2 on Conjugate Gradient Methods: Mrinmay Maharaj

[3] Numerical Optimization: Jorge Nocedal, Stephen J. Wright

[4] Testing Different Conjugate Gradient Methods For Large Scale Unconstrained Optimization: Yu-hong Dai, Qin Ni

[5] A modified PRP conjugate gradient method: Gonglin Yuan, Xiwen Lu

[6] Preconditioning for Hessian-Free Optimization: Robert Seidl

[7] Anonymous write up on Preconditioning from Illinois Institute of Technology

[8] Wikipedia Entry on Conjugate Gradient Method

[9] Function minimization by conjugate gradients: R. Fletcher, C. M. Reeves

[10] Numerical Analysis pp 122-141: M. J. D. Powell

[1] Preconditioned nonlinear conjugate gradient method for micromagnetic energy minimization: Lukas Exl, Johann Fischbacher, Alexander Kovacs, Harald Oezelt, Markus Gusenbauer, Thomas Schrefl

I am also thankful to the information on conjugate gradient methods at Netlib and at An Introduction to the Conjugate Gradient Method Without the Agonizing Pain by Jonathan Richard Shewchuk for a humorous and in depth discussion on Conjugate Gradients