Jinchao Xu
Penn State

Finite Element and Deep Learning
基于机器学习的PDE数值计算与应用暑期短课

Summer 2020

Homework 1

Problem 1. Consider the following linear system

$$A * u = f, \tag{1.1}$$

or equivalently $u = \arg\min \frac{1}{2}(A * v, v)_F - (f, v)_F$, where $(f, v)_F = \sum_{i,j=1}^{n} f_{i,j} v_{i,j}$ is the Frobenius inner product. Here $*$ represents a convolution with one channel, stride one and zero padding one. The convolution kernel A is given by

$$A = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix},$$

the solution $u \in \mathbb{R}^{n\times n}$, and the RHS $f \in \mathbb{R}^{n\times n}$ is given by $f_{i,j} = \dfrac{1}{(n+1)^2}$.

Tasks:
Set $J = 4$, $n = 2^J - 1$ and the number of iterations $M = 100$. Use the gradient descent method and the multigrid method to solve the above problem with a random initial guess u^0. Let u_{GD} and u_{MG} denote the solutions obtained by gradient descent and multigrid respectively.

1. Plot the surface of solution u_{GD} and u_{MG}.
2. Define error $e^m_{GD} = \|A * u^m_{GD} - f\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} |(A * u^m_{GD} - f)_{i,j}|^2}$ for $m = 0, 1, 2, 3, ..., M$. Similarly, we define the multigrid error e^m_{MG}. Plot the errors e^m_{GD} and e^m_{MG} as a function of the iteration m (your x-axis is m and your y-axis is the error). Put both plots together in the same figure.
3. Find the minimal m_1 for which $e^{m_1}_{GD} < 10^{-5}$ and the minimal m_2 for which $e^{m_2}_{MG} < 10^{-5}$, and report the computational time for each method. Note that m_1 or m_2 may be greater than $M = 100$, in this case you will have to run more iterations.

Remark:

1. For gradient descent method with $\eta = \frac{1}{8}$, you need to write a code: Given initial guess u^0

$$
\begin{aligned}
&\text{for } m = 1, 2, ..., M \\
&\quad \text{for } i, j = 1 : n \\
&\qquad u^m_{i,j} = u^{m-1}_{i,j} + \eta(f_{i,j} - (A * u^{m-1})_{i,j}) \\
&\quad \text{endfor} \\
&\text{endfor}
\end{aligned}
$$

2. For Multigrid: Given initial guess u^0

$$
\begin{aligned}
&\text{for } m = 1, 2, \cdots \text{till convergence} \\
&\qquad u^m = \mathrm{MG1}(b; u^{m-1}; J, \nu_1, \cdots, \nu_J). \\
&\text{endfor}
\end{aligned}
$$

   Here, MG1 is described in Algorithm 1.

Algorithm 1 $\mu = \mathrm{MG1}(f; \mu^0; J, \nu_1, \cdots, \nu_J)$

Set up

$$f^1 = b, \quad \mu^1 = \mu^0.$$

Smoothing and restriction from fine to coarse level (nested)

for $\ell = 1 : J$ **do**

  for $i = 1 : \nu_\ell$ **do**

$$\mu^\ell \leftarrow \mu^\ell + S^\ell * (f^\ell - A_\ell * \mu^\ell).$$

  end for

  Form restricted residual and set initial guess:

$$\mu^{\ell+1} \leftarrow \Pi_\ell^{\ell+1} \mu^\ell, \quad f^{\ell+1} \leftarrow R *_2 (f^\ell - A_\ell * \mu^\ell) + A_{\ell+1} * \mu^{\ell+1},$$

end for

Prolongation and restriction from coarse to fine level

for $\ell = J - 1 : 1$ **do**

$$\mu^\ell \leftarrow \mu^\ell + R *_2^\top (\mu^{\ell+1} - \Pi_\ell^{\ell+1} \mu^\ell).$$

end for

$$\mu \leftarrow \mu^1.$$