APPENDIX

Algorithm 1 **Approximate sparse relation matrix of subdomain – level augmented matrix**

Input result of KL expansion $\mathbf{K}^{s}_{i,II}$ for $i = 0, 1, 2, \cdots, m$ and $s = 1, 2, \cdots, N_s$
for $s = 1, 2, \cdots, N_s$ **do**
  for $i = 0, 1, 2, \cdots, m$ **do**
    Calculate $\mathbf{R}^{s}_{i,II} = (\mathbf{K}^{s}_{0,II})^{-1}\mathbf{K}^{s}_{i,II}$
    Calculate $\tilde{\mathbf{R}}^{s}_{i,II} = \ell\left(\mathbf{R}^{s}_{i,II}\right)$ by (17)
  end for
  Calculate $\tilde{\mathbf{R}}^{s}_{II} = \sum_{i=0}^{m} \Theta\left(\mathbf{A}_i, \tilde{\mathbf{R}}^{s}_{i,II}\right)$
end for
Output $\tilde{\mathbf{R}}^{s}_{II}$

Algorithm 2 **Multiplication of argument matrix and vector**

Input $\mathbf{p}$
Initialize $\mathbf{q}_j = \mathbf{0}$, $j = 0, 1, 2, \ldots, M-1$
for $\left(i, j, k, c_{ijk}\right) \in \tilde{\lambda}$ **do**
  $\mathbf{q}_j = \mathbf{q}_j + \mathbf{K}_i \mathbf{p}_k$
  $\mathbf{q} = \{\ \mathbf{q}_0 \quad \mathbf{q}_1 \quad \cdots \quad \mathbf{q}_{M-1}\ \}$
end for
Output $\mathbf{q}$

Algorithm 3 **Parallelmultiplicationof e – SC matrixandvector**

Input $\mathbf{p}$
for $s = 1, 2, 3, \cdots, N_s$, parallelly **do**
  Compute $\mathbf{p}_s = \mathbf{B}^{s}_{S}\mathbf{p}$
  Compute $\mathbf{q}^{s}_{1} = \mathbf{K}^{s}_{\Gamma\Gamma}\mathbf{p}^{s}$ by Algorism 2
  Compute $\mathbf{q}^{s}_{a} = \mathbf{K}^{s}_{I\Gamma}\mathbf{p}^{s}$ by Algorism 2
  Solve $\mathbf{K}^{s}_{II}\mathbf{q}^{s}_{b} = \mathbf{q}^{s}_{a}$. by PCG, preconditioner $\mathbf{M}^{s} = \left(\mathbf{I} \otimes \mathbf{K}^{s}_{0,II}\right)\tilde{\mathbf{R}}^{s}_{II}$
  Compute $\mathbf{q}^{s}_{2} = \mathbf{K}^{s}_{\Gamma I}\mathbf{q}^{s}_{b}$. by Algorism 2
  Compute $\mathbf{q}^{s} = \mathbf{q}^{s}_{1} - \mathbf{q}^{s}_{2}$
end for
Compute $\mathbf{q} = \sum_{s=1}^{N_s} \left(\mathbf{B}^{s}_{S}\right)^{T} \mathbf{q}^{s}$
Output $\mathbf{q}$

Algorithm 4 Parallel computation for the relation matrix of e – SC matrix

Establish the second level Boolean matrix $\mathbf{B}_L^{(i)}$, for $i = 1, 2, \cdots, L_i$
Compute $\mathbf{S} = \sum_{i=1}^{L_i} \left(\overline{\mathbf{K}}_{0,\Gamma\Gamma}^i - \overline{\mathbf{K}}_{0,\Gamma I}^i \left(\overline{\mathbf{K}}_{0,II}^i \right)^{-1} \overline{\mathbf{K}}_{0,I\Gamma}^i \right)$
Compute $\mathbf{S}^{-1}$
for $i = 0, 1, 2, \cdots$, parallelly **do**
  Choose j and k satisfy $(i, j, k) \in \mathfrak{J}$
  Compute $\mathbf{R}_{jk}^{(L_i+1, L_i+1)}$ by Equation (47)
  for $s = 1, 2, 3, \cdots, L_s$ **do**
    Compute $\mathbf{R}_{jk}^{(L_i+1, s)}$ by Equation (45)
    Compute $\mathbf{R}_{jk}^{(s, L_i+1)}$ by Equation (48)
    for $t = 0, 1, 2, 3, \cdots, L_s$ **do**
      Compute $\mathbf{R}_{jk}^{(t,s)}$ by Equation (46)
      Approximate $\tilde{\mathbf{R}}_{jk}^{(t,s)} = \ell\left(\mathbf{R}_{jk}^{(t,s)}\right)$ by Equation (17)
    end for
    Approximate $\tilde{\mathbf{R}}_{jk}^{(L_i+1,s)} = \ell\left(\mathbf{R}_{jk}^{(L_i+1,s)}\right)$ by Equation (17)
    Approximate $\tilde{\mathbf{R}}_{jk}^{(s,L_i+1)} = \ell\left(\mathbf{R}_{jk}^{(s,L_i+1)}\right)$ by Equation (17)
    Approximate $\tilde{\mathbf{R}}_{jk}^{(L_i+1,L_i+1)} = \ell\left(\mathbf{R}_{jk}^{(L_i+1,L_i+1)}\right)$ by Equation (17)
    Assembly $\tilde{\mathbf{R}}'_{jk}$ by Equation (49)
    Approximate $\tilde{\mathbf{R}}_i = \ell(\mathbf{R}_{jk})$ by Equation (17)
  end for
end for
Assembly $\tilde{\mathbf{R}}$ according to $\tilde{\mathbf{R}}_{jk}$ by $\sum_{i=0}^{m} c_{irq}\mathbf{R}_i$
Output $\tilde{\mathbf{R}}$

Algorithm 5 Approximate sparse approach for the e – SC system

Input $\mathbf{u}_\Gamma = 0; \varepsilon = 1$
Parallelly compute $\mathbf{r} = \mathbf{g} - \overline{\mathbf{K}}\mathbf{u}_\Gamma$ by Algorism 3
Parallel Preconditioned Residual: $\mathbf{r} = \Theta\left(\mathbf{I}, \overline{\mathbf{K}}_0\right)\tilde{\mathbf{R}}\mathbf{z}$
Compute: $\mathbf{p} = \mathbf{z}$
Compute: $\delta = (\mathbf{r}, \mathbf{z})$
While $\varepsilon \leq 10^{-8}$, **do** :
Parallelly compute $\mathbf{q} = \overline{\mathbf{K}}\mathbf{p}$ by Algorism 3
Compute: $\gamma = (\mathbf{q}, \mathbf{p})$
Compute: $\alpha = \delta/\gamma$
Update:$\mathbf{u}_\Gamma = \mathbf{u}_\Gamma + \alpha\mathbf{p}$
Update: $\mathbf{r} = \mathbf{r} - \alpha\mathbf{q}$
Parallel Preconditioned Residual: $\mathbf{r} = \Theta\left(\mathbf{I}, \overline{\mathbf{K}}_0\right)\tilde{\mathbf{R}}\mathbf{z}$
Compute: $\beta = (\mathbf{r}, \mathbf{z})/\delta$
Compute: $\delta = \beta\delta$
Update: $\mathbf{p} = z + \beta\mathbf{p}$
$\varepsilon = (\mathbf{r}, \mathbf{r})/(\mathbf{g}, \mathbf{g})$
End while
Output : $\mathbf{u}_\Gamma$