In this chapter we shall discuss subspace correction method and auxiliary space method developed by Xu [5, 7, 8] on solving the linear operator equation
\[ Au = f, \]
posed on a finite dimensional Hilbert space \( \mathbb{V} \cong \mathbb{R}^N \) equipped with an inner product \((\cdot, \cdot)\).
Here \( A : \mathbb{V} \mapsto \mathbb{V} \) is an symmetric and positive definite (SPD) operator, \( f \in \mathbb{V} \) is given, and we are looking for \( u \in \mathbb{V} \) such that (1) holds. How to solve (1) efficiently remains a basic question in numerical PDEs (and in all scientific computing).

1. Space Decomposition and Subspace Correction Methods

In the spirit of dividing and conquering, we decompose the space \( \mathbb{V} \) as the summation of subspaces and correspondingly decompose the problem (1) into sub-problems with smaller size which are relatively easy to solve. This method is developed by Xu [5].

Let \( \mathbb{V}_i \subset \mathbb{V}, i = 1, \ldots, J \), be subspaces of \( \mathbb{V} \). If \( \mathbb{V} = \sum_{i=1}^{J} \mathbb{V}_i \), then \( \{\mathbb{V}_i\} \) is called a space decomposition of \( \mathbb{V} \). By the definition, for any \( u \in \mathbb{V} \), we can decompose \( u \) as
\[ u = \sum_{i=1}^{J} u_i, \quad u_i \in \mathbb{V}_i, \quad i = 1, \ldots, J. \]

Since \( \sum_{i=1}^{J} \mathbb{V}_i \) is not necessarily a direct sum, decompositions of \( u \in \mathbb{V} \) of the form \( u = \sum_{i=1}^{J} u_i \) are in general not unique. We recall some background on the sum of linear spaces in the appendix.

We introduce the following operators for \( i = 1, 1, \ldots, J \):
- \( I_i : \mathbb{V}_i \mapsto \mathbb{V} \) the natural inclusion;
- \( Q_i : \mathbb{V} \mapsto \mathbb{V}_i \) the projection in the inner product \((\cdot, \cdot)\);
- \( P_i : \mathbb{V} \mapsto \mathbb{V}_i \) the projection in the inner product \((\cdot, \cdot)_A\);
- \( A_i : \mathbb{V}_i \mapsto \mathbb{V}_i \) the restriction of \( A \) on the subspace \( \mathbb{V}_i \times \mathbb{V}_i \);
- \( R_i : \mathbb{V}_i \mapsto \mathbb{V}_i \) an approximation of \( A_i^{-1} \) which is often known as smoothers or local subspace solvers.
- \( T_i : \mathbb{V} \mapsto \mathbb{V}_i \quad T_i = R_i Q_i A = R_i A_i P_i. \)

We then explore relations between these operators. By definition
\[ (Q_i^T v_i, v) = (v_i, Q_i v) = (I_i v_i, v) = (I_i v_i, v) \quad \forall v_i \in \mathbb{V}_i, \quad v \in \mathbb{V}, \]
therefore \( Q_i^T \) coincides with the natural inclusion \( I_i \) or equivalently \( I_i^T = Q_i \). In the continuous level, \( I_i \) is the identity operator and thus skipped in many places. In the implementation, the prolongation matrix is the representation of \( I_i \) relative to certain bases and the transpose \( I_i^T \) is the restriction matrix. The matrix or operator \( A \) is understood as the bilinear function on \( \mathbb{V} \times \mathbb{V} \). Then the restriction on subspaces is \( A_i = I_i^T A I_i \).

It follows from the definition that
\[ A_i P_i = Q_i A, \]
namely the following diagram is commutative
\[
\begin{array}{c}
\mathbb{V} \xrightarrow{A} \mathbb{V} \\
\downarrow P_i \quad \downarrow Q_i \\
\mathbb{V}_i \xrightarrow{A_i} \mathbb{V}_i
\end{array}
\]

The consistent notation for the smoother \( R_i \) is \( B_i \), the iterator for each local problem. But we reserve the notation \( B \) for the iterator of the original problem.

Last, let us look at \( T_i = R_iQ_iA = R_iA_iP_i \). When \( R_i = A_i^{-1} \), from the definition, \( T_i = P_i = A_i^{-1}Q_iA \). When \( T_i|_{\mathbb{V}_i} : \mathbb{V}_i \rightarrow \mathbb{V}_i \), the projection \( P_i \) is identity and thus \( T_i|_{\mathbb{V}_i} = R_iA_i \). With a slight abuse of notation, we use \( T_i^{-1} = (T_i|_{\mathbb{V}_i})^{-1} \). The action of \( T_i \) and \( T_i^{-1} \) is
\[
(T_iu_i, u_i)_A = (R_iA_iu_i, A_iu_i), \quad (T_i^{-1}u, u)_A = (R_i^{-1}u, u).
\]

Now we describe the method of subspace correction. For a given residual \( r \), let \( r_i = Q_ir \) denote the restriction of the residual to the subspace, we shall solve the residual equation in the subspaces
\[
A_i e_i^* = r_i \quad \text{by} \quad e_i = R_i r_i.
\]
Subspace corrections \( e_i \) are assembled to give a correction in the space \( \mathbb{V} \) and therefore the method is called subspace correction method.

Basically there are two ways to assemble subspace corrections.

**Parallel Subspace Correction (PSC).** This method is to do the correction on each subspace in parallel. In operator form, it is
\[
u_{k+1} = u_k + B_a(f - A u_k) = (I - B_a A)u_k + B_a f,
\]
where
\[
B_a = \sum_{i=1}^{J} I_i R_i I_i^T.
\]

The subspace correction is \( \tilde{e}_i = I_i R_i I_i^T (f - A u_k) \), and the correction in \( \mathbb{V} \) is \( \tilde{e} = \sum_{i=1}^{J} \tilde{e}_i \).

**Successive Subspace Correction (SSC).** This method is to do the correction in a successive way. In operator form, it reads
\[
\begin{align*}
u^1 &= u_k, \\
u^{i+1} &= \nu^i + I_i R_i I_i^T (f - A \nu^i), \quad i = 1, \ldots, J, \\
u_{k+1} &= \nu^{J+1}.
\end{align*}
\]
The iterator \( B_m \) is not easy to formulate.

\[
\begin{array}{cccc}
\uparrow r & \rightarrow & \mathbb{V}_1 & \searrow e \\
\swarrow \mathbb{V}_2 & \rightarrow & \mathbb{V}_3 & \nearrow e \\
& \rightarrow \mathbb{V}_1 \rightarrow e \rightarrow \mathbb{V}_2 \rightarrow e \rightarrow \mathbb{V}_3 \rightarrow e.
\end{array}
\]

(a) PSC  \hspace{1cm} (b) SSC

**Figure 1.** Illustration of PSC and SSC.

We have the following error formula for PSC and SSC.
• Parallel Subspace Correction (PSC):
\[ u - u_{k+1} = \left[ I - \sum_{i=1}^{J} T_i \right] (u - u_k); \]

• Successive Subspace Correction (SSC):
\[ u - u_{k+1} = \left[ \prod_{i=1}^{J} (I - T_i) \right] (u - u_k). \]

Thus PSC is also called additive methods while SSC is called multiplicative method. In the notation \( \prod_{i=1}^{J} a_i \), we assume there is a build-in ordering from \( i = 1 \) to \( N \) i.e. \( \prod_{i=1}^{J} a_i = a_0 a_1 \ldots a_N \).

We present algorithms of PSC and SSC in the following form to emphasis it is a procedure to solve the residual equation, i.e., given a residual \( r \), return a correction \( e \). One iteration of PSC or SSC can be used as a preconditioner in PCG.

```
function e = PSC(r)
% Solve the residual equation Ae = r by PSC method
for i = 1:J
    ri = Ii'*r;  % restrict the residual to subspace
    ei = Ri*ri;  % solve the residual equation in subspace
    e = e + Ii*ei;  % prolongate the correction to the big space
end

function e = SSC(r)
% Solve the residual equation Ae = r by SSC method
rd = r;
for i = 1:J
    ri = Ii'*rd;  % restrict the residual to subspace
    ei = Ri*ri;  % solve the residual equation in subspace
    e = e + Ii*ei;  % prolongate the correction to the big space
    rd = r - A*e;  % update residual
end
```

Comparing the above PSC and SSC functions, one can immediately see that in SSC, the residual is updated for each subspace correction while not in PSC. In terms of rate of convergence, SSC is superior. On the other hand, PSC is embarrassingly parallel while SSC is essentially a sequential method.

**Example 1.1.** Let us consider the matrix equation
\[ Au = f, \]
where \( A \) is an \( N \times N \) SPD matrix. Let us take the trivial decomposition of \( \mathbb{R}^N = \sum_{i=1}^{N} \text{span}\{ e_i \} \), where \( \{ e_i, i = 1, \ldots, N \} \) is the canonical basis of \( \mathbb{R}^N \). Then

• for \( R_i = \omega I \), PSC is Richardson method;
• for \( R_i = A_i^{-1} \), PSC is Jacobi method;
• for \( R_i = A_i^{-1} \), SSC is the Gauss-Seidal method.

Later we shall also view PSC as a Jacobi method and SSC as a Gauss-Seidel method for a big system formed in a product space formed by subspaces.
2. Auxiliary Space Methods

In this section, we present a variation of the fictitious space method of Nepomnyaschikh [4] and the auxiliary space method of Xu [6]. We follow the presentation in [1].

Let \( \mathcal{V} \) and \( \mathcal{V} \) be two Hilbert spaces and let \( \Pi : \mathcal{V} \rightarrow \mathcal{V} \) be a surjective map. Denoted by \( \Pi^\dagger : \mathcal{V} \rightarrow \mathcal{V} \) the adjoint of \( \Pi \) in the default inner products
\[ (\Pi^\dagger u, \tilde{v}) = (u, \Pi \tilde{v}) \quad \text{for all } u \in \mathcal{V}, \tilde{v} \in \mathcal{V}. \]

Here, to save notation, we use \((\cdot, \cdot)\) for inner products in both \( \mathcal{V} \) and \( \mathcal{V} \). Since \( \Pi \) is surjective, its transpose \( \Pi^\dagger \) is injective.

Given an SPD operator \( A : \mathcal{V} \rightarrow \mathcal{V} \), let \( \tilde{A} = \Pi^\dagger A \Pi : \tilde{\mathcal{V}} \rightarrow \tilde{\mathcal{V}} \) be the lift of \( A \). To construct a good approximation of \( A^{-1} \), we can project one for \( \tilde{A} \). Let \( \tilde{B} : \tilde{\mathcal{V}} \rightarrow \tilde{\mathcal{V}} \) be SPD, we can define \( B := \Pi \Pi^\dagger : \mathcal{V} \rightarrow \mathcal{V} \) and will show it is also SPD. The relation is summarized in the following diagram

\[
\begin{array}{c}
\mathcal{V} \\
\downarrow \Pi \\
\tilde{\mathcal{V}} \\
\uparrow \Pi^\dagger \\
\mathcal{V}
\end{array}
\]

\[ \tilde{A} \]

\[ \begin{array}{c}
\mathcal{V} \\
\downarrow \Pi \\
\tilde{\mathcal{V}} \\
\uparrow \Pi^\dagger \\
\mathcal{V}
\end{array}
\]

\[ B \]

\[ A \]

\[ \begin{array}{c}
\mathcal{V} \\
\downarrow \Pi \\
\tilde{\mathcal{V}} \\
\uparrow \Pi^\dagger \\
\mathcal{V}
\end{array}
\]

Theorem 2.1. Let \( \mathcal{V} \) and \( \mathcal{V} \) be two Hilbert spaces and let \( \Pi : \tilde{\mathcal{V}} \rightarrow \mathcal{V} \) be a surjective map. Let \( \tilde{B} : \tilde{\mathcal{V}} \rightarrow \tilde{\mathcal{V}} \) be a symmetric and positive definite operator. Then \( B := \Pi \Pi^\dagger : \mathcal{V} \rightarrow \mathcal{V} \) is also symmetric and positive definite. Furthermore
\begin{equation}
(B^{-1}v, v) = \inf_{\Pi \tilde{v} = v} (\tilde{B}^{-1} \tilde{v}, \tilde{v}).
\end{equation}

Proof. We shall adapt the proof given by Xu and Zikatanov [9] (Lemma 2.4).

It is obvious that \( B \) is symmetric and positive semi-definite. Since \( \tilde{B} \) is SPD and \( \Pi^\dagger \) is injective, \( (Bv, v) = (\Pi \Pi^\dagger v, \Pi \Pi^\dagger v) = 0 \) implies \( \Pi^\dagger v = 0 \) and \( v = 0 \). Therefore \( B \) is positive definite.

Let \( \tilde{\tilde{v}} = \tilde{B} \Pi^\dagger B^{-1}v \). Then \( \Pi \tilde{\tilde{v}} = v \). For any \( \tilde{\tilde{u}}, \tilde{\tilde{v}} \in \tilde{\mathcal{V}} \)
\[ (\tilde{B}^{-1} \tilde{\tilde{v}}^*, \tilde{\tilde{u}}) = (\Pi \Pi^\dagger B^{-1}v, \tilde{\tilde{u}}) = (B^{-1}v, \Pi \tilde{\tilde{u}}). \]

In particular
\[ (\tilde{B}^{-1} \tilde{\tilde{v}}^*, \tilde{\tilde{v}}^*) = (B^{-1}v, \Pi \tilde{\tilde{v}}^*) = (B^{-1}v, v). \]

For any \( \tilde{\tilde{v}} \in \tilde{\mathcal{V}} \), denoted by \( v = \Pi \tilde{\tilde{v}} \), we write \( \tilde{v} = \tilde{v}^* + \tilde{\tilde{v}} \) with \( \Pi \tilde{\tilde{v}} = 0 \). Then
\[
\inf_{\Pi \tilde{\tilde{v}} = v} (B^{-1} \tilde{\tilde{v}}, \tilde{\tilde{v}}) = \inf_{\Pi \tilde{\tilde{v}} = 0} (B^{-1}(\tilde{v}^* + \tilde{\tilde{v}}), \tilde{v}^* + \tilde{\tilde{v}})
\]
\[
= (B^{-1}v, v) + \inf_{\Pi \tilde{\tilde{v}} = 0} \left( 2(B^{-1} \tilde{v}^*, \tilde{\tilde{v}}) + (\tilde{B}^{-1} \tilde{\tilde{v}}, \tilde{\tilde{v}}) \right)
\]
\[
= (B^{-1}v, v) + \inf_{\Pi \tilde{\tilde{v}} = 0} (B^{-1} \tilde{\tilde{v}}, \tilde{\tilde{v}})
\]
\[
= (B^{-1}v, v). \]

The symmetric positive definite operator \( B \) can be used as a preconditioner for solving \( Au = f \) using PCG. To estimate the condition number \( \kappa(BA) \), we only need to compare \( B^{-1} \) and \( A \).

Lemma 2.2. For two SPD operators \( A \) and \( B \), if \( c_0(Av, v) \leq (B^{-1}v, v) \leq c_1(Av, v) \) for all \( v \in \mathcal{V} \), then \( \kappa(BA) \leq c_1/c_0 \).
Proof. Note that $BA$ is symmetric with respect to $A$. Therefore
\[
\lambda_{\min}^{-1}(BA) = \lambda_{\max}((BA)^{-1}) = \sup_{u \in \mathcal{V} \setminus \{0\}} \left( (BA)^{-1}u, u \right)_A = \sup_{u \in \mathcal{V} \setminus \{0\}} \left( B^{-1}u, u \right).
\]
Therefore $(B^{-1}v, v) \leq c_1(Av, v)$ implies $\lambda_{\min}(BA) \geq c_1^{-1}$. Similarly $(B^{-1}v, v) \geq c_0(Av, v)$ implies $\lambda_{\max}(BA) \leq c_0^{-1}$. The estimate of $\kappa(BA)$ then follows. \qed

**Theorem 2.3.** Let $\mathcal{V}$ and $\mathcal{V}$ be two Hilbert spaces and let $\Pi : \mathcal{V} \to \mathcal{V}$ be a symmetric and positive definite operator and $B = \Pi B \Pi^\top$. If
\[
c_0(Av, v) \leq \inf_{\Pi \tilde{v} = v} (\tilde{B}^{-1} \tilde{v}, \tilde{v}) \leq c_1(Av, v) \quad \text{for all } v \in \mathcal{V},
\]
then
\[
\kappa(BA) \leq c_1/c_0.
\]

**Remark 2.4.** In literature, e.g. the fictitious space lemma of [4], the condition (6) is usually decomposed to the following two conditions:
1. For any $v \in \mathcal{V}$, there exists a $\tilde{v} \in \mathcal{V}$, such that $\Pi \tilde{v} = v$ and $\|\tilde{v}\|_B^{-1} \leq c_1\|v\|_A$.
2. For any $\tilde{v} \in \mathcal{V}$, $\|\Pi \tilde{v}\|_\mathcal{V}^2 \leq c_0^{-1}\|\tilde{v}\|_B^{-1}^2$.

3. **AN AUXILIARY SPACE OF PRODUCT TYPE**

Given a space decomposition $\mathcal{V} = \bigoplus_{i=1}^J \mathcal{V}_i$, we construct an auxiliary space of product type $\tilde{\mathcal{V}} = \mathcal{V}_0 \times \mathcal{V}_1 \times \ldots \times \mathcal{V}_J$, with the standard inner product for product spaces $(\tilde{u}, \tilde{v}) := \sum_{i=1}^J (u_i, v_i)$. We define $\Pi : \tilde{\mathcal{V}} \to \mathcal{V}$ as $\Pi \tilde{u} = \sum_{i=1}^J u_i$. In operator form $\Pi = (I_1, I_2, \ldots, I_J)$ if we treat $\tilde{u} = (u_0, \cdots, u_J)^\top$ as a column vector. Since $\mathcal{V} = \bigoplus_{i=1}^J \mathcal{V}_i$, the operator $\Pi$ is surjective.

Let $\tilde{A} = \Pi \Pi^\top A \Pi$ and $\tilde{f} = \Pi f$. If $\tilde{u}$ is a solution of $\tilde{A} \tilde{u} = \tilde{f}$, by multiplying $\Pi^\top$ both sides, it is straightforward to verify that then $u = \Pi R \tilde{u}$ is the solution of $Au = f$.

We shall derive PSC and SSC by classical iterative methods of solving $\tilde{A} \tilde{u} = \tilde{f}$. To this purpose, let $R_i : \mathcal{V}_i \to \mathcal{V}_i$ be nonsingular operators, often known as smoothers, approximating $A_i^{-1}$. Define a diagonal matrix of operators $\tilde{R} = \text{diag}(R_0, R_1, \cdots, R_J) : \tilde{\mathcal{V}} \to \mathcal{V}$ which is also non-singular.

By direct computation, the entry $\tilde{a}_{ij} = Q_iA_i I_j = A_i Q_j I_j$. In particular $\tilde{a}_{ii} = A_i$. The symmetric operator $\tilde{A}$ may be singular with nontrivial kernel $\text{ker}(\Pi)$, but the diagonal of $\tilde{A}$ is always non-singular. Write $\tilde{A} = \tilde{D} + \tilde{L} + \tilde{U}$ where $\tilde{D} = \text{diag}(A_0, A_1, \cdots, A_J)$, $\tilde{L}$ and $\tilde{U}$ are lower and upper triangular matrix of operators, and $\tilde{L}^\top = \tilde{U}$.

Considering the iteration
\[
\tilde{u}_{k+1} = \tilde{u}_k + \tilde{R}(\tilde{f} - \tilde{A}\tilde{u}_k).
\]
Let $u_k = \Pi \tilde{u}_k$. Applying $\Pi$ to (7) and noting that $\tilde{f} = \Pi^\top f$, and $\tilde{A} \tilde{u}_k = \Pi^\top A u_k$, we obtain the PSC method
\[
u_{k+1} = u_k + \sum_{i=1}^J R_i Q_i (f - A u_k),
\]
The multiplicative method is more subtle. Following [3], we shall view the SSC for solving $Au = f$ as a Gauss-Seidel type method for $\tilde{A} \tilde{u} = \tilde{f}$. 


Lemma 3.1. Let $\tilde{A} = \tilde{D} + \tilde{L} + \tilde{U}$ and $\tilde{B} = (\tilde{R}^{-1} + \tilde{L})^{-1}$. Then SSC for $Au = f$ with local solvers $R_i$ is equivalent to the Gauss-Seidel type method for solving $\tilde{A}u = \tilde{f}$:

\begin{equation}
\tilde{u}_{k+1} = \tilde{u}_k + \tilde{B}(\tilde{f} - \tilde{A}\tilde{u}_k).
\end{equation}

Proof. By multiplying $\tilde{R}^{-1} + \tilde{L}$ to (8) and rearranging the terms, we have

\begin{equation}
\tilde{R}^{-1}\tilde{u}_{k+1} = \tilde{R}^{-1}\tilde{u}_k + \tilde{f} - \tilde{L}\tilde{u}_{k+1} - (\tilde{D} + \tilde{U})\tilde{u}_k.
\end{equation}

Multiplying $\tilde{R}$, we obtain

\begin{equation}
\tilde{u}_{k+1} = \tilde{u}_k + \tilde{R}(\tilde{f} - \tilde{L}\tilde{u}_{k+1} - (\tilde{D} + \tilde{U})\tilde{u}_k),
\end{equation}

and its component-wise formula, for $i = 1, \cdots, J$

\begin{equation}
u_{k+1} = u_k + R_i \left( f_i - \sum_{j=1}^{i-1} \tilde{a}_{ij}u_{k+1}^j - \sum_{j=1}^{J} \tilde{a}_{ij}u_k^j \right)
= u_k + R_i Q_i \left( f - A \sum_{j=1}^{i-1} u_{k+1}^j - \sum_{j=1}^{J} u_k^j \right).
\end{equation}

Let the dynamic update

\begin{equation}
v^i = \Pi (u_{k+1}^i, u_k^i, u_{k+1}^{i+1}, \cdots, u_{k+1}^J, u_k^1, \cdots, u_k^J) \tau = \sum_{j=1}^{i} u_{k+1}^j + \sum_{j=i+1}^{J} u_k^j.
\end{equation}

Noting that $v^i - v^{i-1} = u_{k+1}^i - u_k^i$, we then get, for $i = 1, \cdots, J + 1$

\begin{equation}
v^i = v^{i-1} + R_i Q_i (f - Av^{i-1}),
\end{equation}

which is exactly the correction in the subspace $V_i$; see (4). \hfill \Box

Recall that for SSC, for each subspace problem, we have the operator form $v^{i+1} = v^i + R_i (f - Av^i)$, but it is not easy to write out the iterator for the space $V$. Let us define $B_m$ to be the error operator so that

\begin{equation}
I - B_m A = (I - R_J Q_J A)(I - R_{J-1} Q_{J-1} A) \cdots (I - R_0 Q_0 A).
\end{equation}

We can then derive a formulation of $B_m$ from the auxiliary space method. Let $\tilde{B}_m = (\tilde{R}^{-1} + \tilde{L})^{-1}$ and its symmetrization as

\begin{equation}
\tilde{B}_m = \tilde{B}_m^T + \tilde{B}_m - \tilde{B}_m^T \hat{A} \tilde{B}_m = \tilde{B}_m^T (\tilde{B}_m^{-T} + \tilde{B}_m^{-1} \tilde{B}_m^{-T} \hat{A} \tilde{B}_m).
\end{equation}

Lemma 3.2. For SSC, we have

\begin{equation}
B_m = \Pi \tilde{B}_m \Pi^T \quad \text{and} \quad \bar{B}_m = \Pi \bar{B}_m \Pi^T.
\end{equation}

Proof. Let $u_k = \Pi \tilde{u}_k$. Applying $\Pi$ to (8) and noting that

\begin{equation}
\tilde{f} = \Pi^T f, \quad \text{and} \quad \hat{A} \tilde{u}_k = \Pi^T A u_k,
\end{equation}

we then get

\begin{equation}
u_{k+1} = u_k + \Pi \tilde{B}_m \Pi^T (f - A u_k).
\end{equation}

The formulae for $\bar{B}_m$ follows from a similarly computation. \hfill \Box
4. IDENTITIES FOR ADDITIVE AND MULTIPLICATIVE METHODS

The operator for the additive method is

\[ B_a = \Pi \tilde{\mathbf{R}} \Pi^\top = \sum_{i=1}^{J} I_i R_i I_i^\top. \]

Applying Theorem 2.1, we obtain the following identity for preconditioner \( B_a \).

**Theorem 4.1.** If \( R_i \) is SPD on \( \mathbb{V}_i \) for \( i = 1, \ldots, J \), then \( B_a \) defined by (10) is SPD on \( \mathbb{V} \). Furthermore

\[ (B_a^{-1} v, v) = \inf_{\sum_{i=1}^{J} v_i = v} \sum_{i=1}^{J} \left( (R_i^{-1} v_i, v_i) + \sum_{j=1 \atop j \neq i}^{J} \sum_{v_j} (T_i^{-1} v_i, v_j) \right). \]

To compute \( \tilde{B}_m \), we define the diagonal matrix of operators \( \tilde{\mathbf{R}} = \text{diag}(\tilde{R}_1, \tilde{R}_2, \ldots, \tilde{R}_J) \), where, for each \( R_i, i = 1, \ldots, J \), its symmetrization is

\[ \mathbf{R}_i = R_i^T (R_i^{-1} + R_i^{-1} - A_i) R_i. \]

Substituting \( \tilde{B}_m^{-1} = \tilde{R}^{-1} + \tilde{L} \), and \( \mathbf{A} = \tilde{D} + \tilde{L} + \tilde{U} \) into (9), we have

\[ \tilde{B}_m = (\tilde{R}^T + \tilde{L})^{-1}(\tilde{R}^T + \tilde{R}^{-1} - \tilde{D})(\tilde{R}^{-1} + \tilde{L})^{-1} \]

\[ = (\tilde{R}^{-T} + \tilde{L})^{-1} \tilde{R}^{-T} \tilde{R}^{-1} (\tilde{R}^{-1} + \tilde{L})^{-1}. \]

It is obvious that \( \tilde{B}_m \) is symmetric. To be positive definite, from (13), it suffices to assume \( \tilde{R}_i \), i.e. each \( \mathbf{R}_i, i = 1, \ldots, J \), is symmetric and positive definite which is equivalent to the operator \( I - \mathbf{R}_i A_i \) is a contraction and so is \( I - \mathbf{R}_i A_i \).

(C) \[ \| I - \mathbf{R}_i A_i \|_{A_i} < 1 \] for each \( i = 1, \ldots, J \).

**Theorem 4.2.** Suppose (C) holds. Then \( \tilde{B}_m = \Pi \tilde{B}_m \Pi^\top \) is SPD, and

\[ (\tilde{B}_m^{-1} v, v) = \inf_{\sum_{i=1}^{J} v_i = v} \sum_{i=1}^{J} \| v_i + R_i^T A_i P_i \sum_{j=i+1}^{J} v_j \|_{\mathbf{R}_i^{-1}}^2. \]

\[ (\tilde{B}_m^{-1} v, v) = \| v \|_A^2 + \inf_{\sum_{i=1}^{J} v_i = v} \sum_{i=1}^{J} \| R_i^T (A_i P_i \sum_{j=i+1}^{J} v_j - R_i^{-1} v_i) \|_{\mathbf{R}_i^{-1}}^2. \]

In particular, for \( R_i = A_i^{-1} \), we have

\[ (\tilde{B}_m^{-1} v, v) = \| v \|_A^2 + \inf_{\sum_{i=1}^{J} v_i = v} \sum_{i=1}^{J} \| P_i \sum_{j=i+1}^{J} v_j \|_A^2. \]

**Proof.** From (13), we have

\[ \tilde{B}_m^{-1} = (\tilde{I} + \tilde{R}^T \tilde{U}) \tilde{R}^{-1} (\tilde{I} + \tilde{R}^T \tilde{U}). \]

Using component-wise formula of

\[ (\tilde{U} \tilde{v})_i = \sum_{j=i+1}^{J} \tilde{a}_{ij} v_j = \sum_{j=i+1}^{J} A_i P_i v_j, \]

and Theorem 2.1, we get (14).
Before we prove the identity (15), we first prove the special case (16) to present the main idea. Obviously (C) holds for the exact local solver $R_i = A_i^{-1}$. When $R_i = A_i^{-1}$, $B_m = (\hat{D} + \hat{L})^{-1}$ and, by direct computation,

\begin{align}
(17) \quad \overline{B}_m^{-1} = \hat{A} + \hat{L}\hat{D}^{-1}\hat{U}.
\end{align}

Therefore

\begin{align}
(\overline{B}_m^{-1} v, v) = \inf_{\hat{v} = v} (\overline{B}_m^{-1} \hat{v}, \hat{v}) = (\hat{A}\hat{v}, \hat{v}) + \inf_{\hat{v} = v} (\hat{D}^{-1}\hat{U}\hat{v}, \hat{U}\hat{v}).
\end{align}

For any $\hat{v} \in \hat{V}$, denoted by $v = \Pi\hat{v}$, we have

\begin{align}
(\hat{A}\hat{v}, \hat{v}) = (\Pi^T \Pi \hat{v}, \hat{v}) = (\Pi \hat{v}, \Pi \hat{v}) = \|v\|_A^2,
\end{align}

and

\begin{align}
(\hat{D}^{-1}\hat{U}\hat{v}, \hat{U}\hat{v}) &= \sum_{i=1}^J (A_i^{-1} \sum_{j=i+1}^J A_i P_i v_j, \sum_{j=i+1}^J A_i P_i v_j) = \sum_{i=1}^J \|P_i \sum_{j=i+1}^J v_j\|_A^2.
\end{align}

The identity (16) then follows.

In general let

\begin{align}
\mathcal{M} = \hat{R}^{-T} + \hat{R}^{-1} - \hat{D} = \hat{R}^{-T}\hat{R}\hat{R}^{-1}, \quad \mathcal{U} = \hat{D} + \hat{U} - \hat{R}^{-1}, \quad \mathcal{L} = \mathcal{U}^T.
\end{align}

then $\hat{R}^{-1} + \hat{L} = \mathcal{M} + \mathcal{L}$ and $\hat{A} = \mathcal{M} + \mathcal{U} + \mathcal{L}$. We then compute, from (12), that

\begin{align}
\overline{B}_m^{-1} &= (\hat{R}^{-1} + \hat{L})(\hat{R}^{-T} + \hat{R}^{-1} - \hat{D})^{-1}(\hat{R}^{-T} + \hat{L}^T) \\
&= (\mathcal{M} + \mathcal{L})\mathcal{M}^{-1}(\mathcal{M} + \mathcal{U}), \\
&= \hat{A} + \mathcal{L}\mathcal{M}^{-1}\mathcal{U} \\
&= \hat{A} + \left[\hat{R}^T(\hat{D} + \hat{U} - \hat{R}^{-1})\right]^T \hat{R}^{-1} \left[\hat{R}^T(\hat{D} + \hat{U} - \hat{R}^{-1})\right].
\end{align}

The identity (15) then follows from the component-wise formula. \hfill \Box

If we use the operator $T_i = R_i A_i P_i : \mathcal{V} \to \mathcal{V}_i$, then $T_i^* = R_i^T A_i P_i$, $T_i := T_i + T_i^* - T_i^* T_i = R_i A_i P_i$, and $(R_i^T u_i, u_i) = (T_i^{-1} u_i, u_i)_A$. Here $T_i^{-1} := (T_i|_{\mathcal{V}_i})^{-1} : \mathcal{V}_i \to \mathcal{V}_i$ is well defined due to the assumption (C). The identity (15) can be written as the original formulation in [9]

\begin{align}
(\overline{B}_m^{-1} v, v) = \|v\|_A^2 + \inf_{\sum_{i,v_i} v_i = v} \sum_{i=1}^J (T_i^{-1} T_i^* w_i, T_i^* w_i)_A,
\end{align}

with $w_i = \sum_{j=i}^J v_j - T_i^{-1} v_i$. The identity (14) becomes the formula in [2]

\begin{align}
(\overline{B}_m^{-1} v, v) = \inf_{\sum_{i,v_i} v_i = v} \sum_{i=1}^J (T_i^{-1} (v_i + T_i^* \sum_{k=i+1}^J v_k), v_i + T_i^* \sum_{k=i+1}^J v_k)_A.
\end{align}

Combining Lemma 3.2 and Theorem 4.2, we obtain the X-Z identity for multiplicative methods.

**Theorem 4.3 (X-Z identity).** Suppose assumption (C) holds. Then

\begin{align}
(18) \quad ||I - B_mA||_A^2 = ||I - \overline{B}_mA||_A = 1 - \frac{1}{K}.
\end{align}
In particular, for
\[ R \parallel (19) \]
Or
\[ \| I - B_m A \|^2_A = \| I - B_m A \|_A = 1 - \frac{1}{1 + c_0}, \]
where
\[ c_0 = \sup_{\|v\|_A = 1} \inf_{\sum v_i = 0} \sum_{i=1}^J \| R_i^T A_i P_i \sum_{j=1}^J v_j \|_1^2. \]

In particular, for \( R_i = A_i^{-1} \),
\[ \|(I - P_j)(I - P_{j-1}) \cdots (I - P_0)\|^2_A = 1 - \frac{1}{1 + c_0}, \]
where
\[ c_0 = \sup_{\|v\|_A = 1} \inf_{\sum v_i = 0} \sum_{i=1}^J \| P_i \sum_{j=i+1}^J v_j \|_A^2. \]

APPENDIX: SUM AND PRODUCT OF TWO LINEAR SPACES

Given two linear spaces \( V_1, V_2 \) and assume they are subspaces of a larger linear space \( V \). We have the following operations of these two spaces
- \( V_1 + V_2 = \{ v_1 + v_2 : v_1 \in V_1, v_2 \in V_2 \} \);
- \( V_1 \oplus V_2 = V_1 + V_2 \) and \( V_1 \cap V_2 = \{0\} \);
- \( V_1 \times V_2 = \{(v_1, v_2) : v_1 \in V_1, v_2 \in V_2 \} \);
- \( V_1 \otimes V_2 = \{v_1 \otimes v_2 : v_1 \in V_1, v_2 \in V_2 \} \).

The tensor product \( v_1 \otimes v_2 \) is a bilinear mapping on the dual space \( V_1' \times V_2' \). A natural product topology can be defined for \( V_1 \times V_2 \) component-wise.

The relation of dimensions are
- \( \dim(V_1 + V_2) = \dim(V_1) + \dim(V_2) - \dim(V_1 \cap V_2) \leq \dim(V_1) + \dim(V_2) \);
- \( \dim(V_1 \oplus V_2) = \dim(V_1) + \dim(V_2) \);
- \( \dim(V_1 \times V_2) = \dim(V_1) + \dim(V_2) \);
- \( \dim(V_1 \otimes V_2) = \dim(V_1) \times \dim(V_2) \).

We emphasize the sum \( V_1 + V_2 \) may not enlarge the space. For example, when \( V_1 \subset V_2 \) (a line on a plane), \( V_1 + V_2 = V_2 \).

REFERENCES


