Specifically, we will consider the "preconditioner" defined as:

$$M = M^{-1} = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$$

Recall that PCG involves using the M inner product. For this M, when is $(v, w)_M := (v, Mw)$ a valid inner product?

A valid inner product (u, v) must satisfy 3 conditions:

- $(u, av + bw) = a(u, v) + b(u, w)$ for all u, v, w
- $(u, v) = (v, u)$ for all u, v
- $(u, u) \geq 0$ for all u, and $(u, u) = 0$ if and only if $u = 0$

The linearity of the multiplication operation **Mw** satisfies the first condition. The second condition is satisfied by **M** being symmetric. The third condition isn't satisfied in general for this choice of **M** since it is not positive definite, instead the inner product is valid for only some choice of vectors. Consider a general vector **u**, which we split into upper and lower halves **a** and **b**, and its **M**-inner product:

$$u = \begin{bmatrix} a \\ b \end{bmatrix},\ (u, u)_M = \begin{bmatrix} a\, b \end{bmatrix} \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a\, b \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = (a, b) + (b, a) = 2(a, b)$$

The third condition will hold if $(a, b) > 0$, for non-zero a and b.

In the context of BCG we can see that the **M**-inner product is valid. If we attempt to arrive at BCG via PCG with this choice of **M** we must compute two **M**-inner products: $(z_j, z_j)_M$ and $(M^{-1}Ap_j, p_j)_M$. Both z_j and p_j are computed from r_j, which itself is composed of two halves: the residuals from the primary system and its dual. By the very definition of the bi-orthogonal basis these two halves of the residual are orthogonal two each other, and therefore $(r_j, r_j^*) > 0$, ensuring the **M**-inner product is valid.

Show that the result is the BiCG algorithm.

Consider the substitution of the following into the PCG algorithm (Saad Alg 9.1):

$$B = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix},\ \bar{x} = \begin{bmatrix} x^* \\ x \end{bmatrix},\ \bar{b} = \begin{bmatrix} b \\ b^* \end{bmatrix},\ B\bar{x} = \bar{b},\ \bar{z} = M^{-1}\bar{r} = \begin{bmatrix} r^* \\ r \end{bmatrix},\ \bar{p} = \begin{bmatrix} p^* \\ p \end{bmatrix}$$

ALGORITHM **9.1** *Preconditioned Conjugate Gradient*

Compute $r_0 := b - Ax_0$, $z_0 = M^{-1}r_0$, *and* $p_0 := z_0$
For $j = 0, 1, \ldots,$ *until convergence Do:*
$\alpha_j := (r_j, z_j)/(Ap_j, p_j)$
$x_{j+1} := x_j + \alpha_j p_j$
$r_{j+1} := r_j - \alpha_j A p_j$
$z_{j+1} := M^{-1} r_{j+1}$
$\beta_j := (r_{j+1}, z_{j+1})/(r_j, z_j)$
$p_{j+1} := z_{j+1} + \beta_j p_j$
EndDo

ALGORITHM **7.3** *Biconjugate Gradient (BCG)*

Compute $r_0 := b - Ax_0$. *Choose* r_0^* *such that* $(r_0, r_0^*) \neq 0$.
Set, $p_0 := r_0$, $p_0^* := r_0^*$
For $j = 0, 1, \ldots,$ *until convergence Do:*
$\alpha_j := (r_j, r_j^*)/(Ap_j, p_j^*)$
$x_{j+1} := x_j + \alpha_j p_j$
$r_{j+1} := r_j - \alpha_j A p_j$
$r_{j+1}^* := r_j^* - \alpha_j A^T p_j^*$
$\beta_j := (r_{j+1}, r_{j+1}^*)/(r_j, r_j^*)$
$p_{j+1} := r_{j+1} + \beta_j p_j$
$p_{j+1}^* := r_{j+1}^* + \beta_j p_j^*$
EndDo

Figure 1: Comparison between PCG and BCG algorithms. [Saad, IMfSLS]

If we were to solve the full system B we would receive both the solution vector x to the problem $Ax = b$, as well as the solution to the dual system, x^*. Assuming we are just interested in the solution to the primary system we don't need to worry about updating x_j^*. Starting with Line 3 of Alg 9.1:

$$\alpha_j = (\begin{bmatrix} r \\ r^* \end{bmatrix}, \begin{bmatrix} r^* \\ r \end{bmatrix})/(\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} p^* \\ p \end{bmatrix}, \begin{bmatrix} p^* \\ p \end{bmatrix}) \rightarrow (r, r^*)/(Ap, p^*)$$

where we considered only the first row, associated with the primary system.

Lines 4 and 5 remain unchanged, the p_j associated with the α_j calculated in Line 3 should be used. We also need to add a line to compute r_{j+1}^* alongside of computing r_{j+1}; although we don't need to retain the solution vector to the dual system the residual is still needed for the bi-orthogonalization. Line 6 is subsumed into Lines 7 and 8 thanks to the simple substitution available for $z_{j+1} = r_j^*$. Next, Lines 7 and 8 receive the substitution for the old and new z. Finally, we also need to calculate the new p_{j+1}^* in addition to the p_{j+1} term.

If one follows through with the execution of the PCG algorithm with the given matrix **B**, "preconditioner" **M**, and the aforementioned substitutions we arrive at the very same algorithm as presented in Saad's Algorithm 7.3, the Biconjugate Gradient method, as depicted alongside PCG in Figure 1.