Problem 2A

Start with the algorithm 10.3.1 for PCG. Consider the step $p_k = z_{k-1} - \beta_k p_{k-1}$. Using the $A$-conjugacy of the $p$'s, $p_k^TAz_{k-1} = \beta_k(p_k^TAp_{k-1})$. Further $p_kAp_k = p_kAz_{k-1} = z_{k-1}A z_{k-1} - \beta_k^2(p_k^TAp_{k-1})$ using the previous two statements. Define:

$$
\omega_k := \frac{z_{k-1}^T A z_{k-1}}{p_k^T Ap_k} = \frac{z_{k-1}^T A z_{k-1}}{z_{k-1}^T A z_{k-1} - \beta_k^2(p_k^T Ap_{k-1})} = \left(1 - \beta_k^2 \frac{p_k^T Ap_{k-1}}{z_{k-1}^T A z_{k-1}}\right)^{-1}
$$

Now consider the step that updates the solution in 10.3.1:

$$
\omega_k = 1 - \gamma_k - 1
$$

where $\gamma_k := \frac{\alpha_k}{\omega_k}$. Define:

$$
\gamma_{k-1} := \frac{z_{k-1}^T A z_{k-1}}{z_{k-1}^T A z_{k-2} \omega_{k-1}}
$$

Note that the definition of $\gamma_{k-1}$ and the recursion for $\omega_k$ are exactly as in algorithm 10.3.10. Also since $p_1 = z_0$ in 10.3.1, this leads to $\omega_1 = 1$ in 10.3.10.

Now consider the step that updates the solution in 10.3.1: $x_k = x_{k-1} + \alpha_k p_k$ with $\alpha_k := r_k^T z_{k-1} / p_k^T Ap_k$. The expression in 10.3.1 seems to be incorrect with $r_k^T z_{k-1} / p_k^T Ap_k$ printed instead of $r_k^T z_{k-1}$. Note that this leads to $\alpha_k = \omega_k \gamma_k z_{k-1}$.

Invoking the previous iteration, we can rewrite this as $x_k = x_{k-2} + \alpha_k p_{k-1} + \alpha_k p_k$. Thus:

$$
x_k - x_{k-2} = \omega_k \gamma_k z_{k-1} + \alpha_k p_{k-1} + \alpha_k p_k
$$

There are many definitions of $\omega_k$ and $\gamma_k$ as above, the update step of algorithm 10.3.10 namely

$$
x_k = x_{k-2} + \omega_k(\gamma_k z_{k-1} + x_{k-1} - x_{k-2})
$$

is simply a combination of the $(k-1)^{th}$ and $k^{th}$ update steps of 10.3.1 i.e. $x_k = x_{k-1} + \alpha_k p_k$. We need to verify the base case - for $k = 1$, the above becomes $x_1 = x_{-1} + \omega_1(\gamma_0 z_0 + x_0 - x_{-1}) = x_0 + \alpha_1 p_1$ since $x_{-1} = 0$, $p_1 = z_0$, $\omega_1 = 1$ and $\alpha_1 = \omega_1 \gamma_0$. Thus the $k = 1$ iteration is identical for 10.3.1 and 10.3.10.

Thus 10.3.10 is completely equivalent to 10.3.1.