If $\nabla J_{u_{k+1}} = 0$, then the algorihm terminates with $u = u_{k+1}$.

As we showed before, the algorithm terminates in at most n iterations.

Example 49.3. Let us take the example of Section 49.6 and apply the conjugate gradient procedure. Recall that

$$\begin{aligned} J(x,y) &= \frac{1}{2}\begin{pmatrix} x & y \end{pmatrix}\begin{pmatrix} 3 & 2 \\ 2 & 6 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} 2 & -8 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \\ &= \frac{3}{2}x^2 + 2xy + 3y^2 - 2x + 8y. \end{aligned}$$

Note that $\nabla J_v = (3x + 2y - 2, 2x + 6y + 8)$,
Initialize the procedure by setting

$$u_0 = (-2, -2), \qquad d_0 = \nabla J_{u_0} = (-12, -8)$$

Step 1 involves calculating

$$\begin{aligned} \rho_0 &= \frac{\langle \nabla J_{u_0}, d_0 \rangle}{\langle A d_0, d_0 \rangle} = \frac{13}{75} \\ u_1 &= u_0 - \rho_0 d_0 = (-2, -2) - \frac{13}{75}(-12, -8) = \left(\frac{2}{25}, -\frac{46}{75}\right) \\ d_1 &= \nabla J_{u_1} + \frac{||\nabla J_{u_1}||^2}{||\nabla J_{u_0}||^2} d_0 = \left(-\frac{2912}{625}, \frac{18928}{5625}\right). \end{aligned}$$

Observe that ρ_0 and u_1 are precisely the *same* as in the case the case of gradient descent with optimal step size parameter. The difference lies in the calculation of d_1. As we will see, this change will make a *huge* difference in the convergence to the unique minimum $u = (2, -2)$.

We continue with the conjugate gradient procedure and calculate Step 2 as

$$\begin{aligned} \rho_1 &= \frac{\langle \nabla J_{u_1}, d_1 \rangle}{\langle A d_1, d_1 \rangle} = \frac{75}{82} \\ u_2 &= u_1 - \rho_1 d_1 = \left(\frac{2}{25}, -\frac{46}{75}\right) - \frac{75}{82}\left(-\frac{2912}{625}, \frac{18928}{5625}\right) = (2, -2) \\ d_2 &= \nabla J_{u_2} + \frac{||\nabla J_{u_2}||^2}{||\nabla J_{u_1}||^2} d_1 = (0, 0). \end{aligned}$$

Since $\nabla J_{u_2} = 0$, the procedure terminates in *two* steps, as opposed to the 31 steps needed for gradient descent with optimal step size parameter.

Hestenes and Stiefel realized that Equations $(*_6)$ can be modified to make the computations more efficient, by having only one evaluation of the matrix A on a vector, namely d_k. The idea is to compute ∇_{u_k} inductively.