Solutions for Problem Set 2.

Problem 1.
Rewrite the left hand side of the inequality in terms of p_k, using the expression $p_k = -B_k^{-1}\nabla f_k$, which implies $\nabla f_k = -B_k p_k$:

$$\cos\theta_k = \frac{-\nabla f_k^T p_k}{\|\nabla f_k\|\,\|p_k\|} = \frac{p_k^T B_k p_k}{\|B_k p_k\|\,\|p_k\|}$$

Since B_k is symmetric and positive definite, it has a full set of real eigenvalues and the corresponding eigenvectors form an orthogonal basis in $\mathcal{R}^n$. Also, for such a matrix the condition number is the ratio of the largest eigenvalue to the smallest one. Since we have to prove the inequality for arbitrary k, we shall suppress the subscript k from now on, to simplify notation.

Denote λ_i, the eigenvalues of B and u_i, the corresponding eigenvectors (by orthogonality we have $u_i^T u_j = \delta_{ij}$, Cronecker's delta). Then $p = \sum \alpha_i u_i$ for some real numbers α_i, which are the coordinates of p in the eigenbasis of B. Then

$$\begin{aligned} p^T B p = \left(\sum_i \alpha_i u_i^T\right) B \left(\sum_j \alpha_j u_j\right) &= \sum_{i,j} \alpha_i\alpha_j u_i^T B u_j = \sum_{i,j} \alpha_i \alpha_j u_i^T \lambda_j u_j = \sum_i \alpha_i^2 \lambda_i \|u_i\|^2 \\ &\geq \lambda_{min} \sum_i \alpha_i^2 \|u_i\|^2 = \lambda_{min}\,\|p\,\|^2 \end{aligned}$$

At the same time,

$$\begin{aligned} \|Bp\,\|\,\|p\,\| = \left\|\sum_i B\alpha_i u_i\right\| \left\|\sum_i \alpha_i u_i\right\| &= \left\|\sum_i \alpha_i \lambda_i u_i\right\| \left\|\sum_i \alpha_i u_i\right\| = \sqrt{\sum_i \alpha_i^2 \lambda_i^2 \|u_i\|^2}\,\sqrt{\sum_i \alpha_i^2\|u_i\|^2} \\ &\leq \lambda_{max}\sqrt{\sum_i \alpha_i^2\|u_i\|^2}\,\sqrt{\sum_i \alpha_i^2\|u_i\|^2} = \lambda_{max}\sum_i \alpha_i^2\|u_i\|^2 = \lambda_{max}\,\|p\,\|^2 \end{aligned}$$

Therefore

$$\cos\theta = \frac{p^T B p}{\|Bp\,\|\,\|p\,\|} \geq \frac{\lambda_{min}\|p\,\|^2}{\lambda_{max}\,\|p\,\|^2} = \frac{\lambda_{min}}{\lambda_{max}} = \frac{1}{\kappa} \geq \frac{1}{M},$$

as required.

Problem 2.
To prove both implications, it is sufficient to establish that there exist constants K_1 and K_2 such that

$$\|(B_k - \nabla^2 f_k)p_k\| \leq K_1\,\|p_k - p_k^N\| \quad \text{and} \quad \|p_k - p_k^N\| \leq K_2\|(B_k - \nabla^2 f_k)p_k\| \tag{1}$$

for k large enough. Indeed, in this case, obviously,

$$\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f_k)p_k\|}{\|p_k\|} \leq K_1 \lim_{k\to\infty} \frac{\|p_k - p_k^N\|}{\|p_k\|}$$

and

$$\lim_{k\to\infty} \frac{\|p_k - p_k^N\|}{\|p_k\|} \leq K_2 \lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f_k)p_k\|}{\|p_k\|},$$

so if one of these limits is zero, the other is necessarily zero as well.

Since $p_k^N = -\nabla^2 f_k^{-1} \nabla f_k = \nabla^2 f_k^{-1} B_k p_k$, we have

$$\|(B_k - \nabla^2 f_k) p_k\| = \|\nabla^2 f_k p_k^N - \nabla^2 f_k p_k\| \leq \|\nabla^2 f_k\| \, \|p_k - p_k^N\|$$

and

$$\|p_k - p_k^N\| = \|(\nabla^2 f_k^{-1} \nabla^2 f_k) p_k - \nabla^2 f_k^{-1} B_k p_k\| \leq \|\nabla^2 f_k^{-1}\| \, \|(B_k - \nabla^2 f_k) p_k\|$$

But if $f \in C^2$ and x_k converges to x^* with $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ positive definite, the norm of the Hessian evaluated at later iterates has to be bounded, so that $\|\nabla^2 f_k\| \leq K_1$ and $\|\nabla^2 f_k^{-1}\| \leq K_2$ as $k \to \infty$, so (1) is proved, and thus so is the desired equivalence.

Problem 3.
First prove two auxilliary identities:

$$r_k^T p_i = 0 \qquad \forall\, i < k \tag{2}$$

and

$$r_j^T r_i = 0 \qquad \forall\, i \neq j \tag{3}$$

We shall use the fact that the residual $r_j \equiv A x_j - b$ for all j; and also the conjugacy of p_k: $p_i^T A p_j = 0$ for $i \neq j$.

$$\begin{aligned} r_k^T p_i = p_i^T r_k = p_i^T\,(A x_k - b) = p_i^T A x_k - p_i^T\,(A x_i - r_i) = p_i^T\,A(x_k - x_i) + p_i^T r_i = \\ = p_i^T\,A(\alpha_i p_i + \,\ldots\, + \alpha_{k-1} p_{k-1}) + p_i^T r_i = p_i^T r_i + \alpha_i p_i^T\,A p_i = 0 \end{aligned}$$

by definition of α_i, so (2) is established. Now

$$r_j^T r_i = r_j^T\,(\beta_i\, p_{i-1} - p_i) = \beta_i r_j^T p_{i-1} - r_j^T p_i = 0$$

by (2), so (3) is also established. The desired identities are now easy to prove.
Since $p_k = -r_k + \beta_k p_{k-1}$, we can left-multiply this expression by r_k^T to get $r_k^T p_k = -r_k^T r_k + \beta_k r_k^T p_{k-1}$. But the second term vanishes by (2), so $r_k^T p_k = -r_k^T r_k$, and thus

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k} = \frac{r_k^T r_k}{p_k^T A p_k}$$

Now take the expression $p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$, left-multiply it by $p_k^T A$ and use conjugacy to get

$$p_k^T A p_{k+1} = 0 = -p_k^T A r_{k+1} + \beta_{k+1} p_k^T A p_k \quad \Rightarrow \quad \beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k} = \alpha_k\,\frac{r_{k+1}^T A p_k}{r_k^T r_k},$$

where we used the alternative representation for α_k that we have already derived. Now, since $x_{k+1} = x_k + \alpha_k p_k$, it follows that $A x_{k+1} = A x_k + \alpha_k A p_k$, and thus $r_{k+1} = r_k + \alpha_k A p_k$. Left-multiplying this identity by r_{k+1}^T and the application of (3) yield $r_{k+1}^T r_{k+1} = \alpha_k r_{k+1}^T A p_k$, so that

$$\beta_{k+1} = \alpha_k\,\frac{r_{k+1}^T A p_k}{r_k^T r_k} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

as required.

Problem 4.
See discussion preceding formula (3.44) in Nocedal and Wright.
The idea of the heuristic is to produce a reasonable first guess for the initial step length at any iteration using the information that we gained about the behavior of $f(x)$ at previous iterations. Once we get this initial guess we can improve upon it using Wolfe or strong Wolfe line search or, when possible, backtrack and sufficient decrease. A good initial guess may be crucial, since making one too large or too small will result in more work per iteration to determine the correct step length, and thus, more cost and longer compuation time.

Here are a couple of possible choices of initial step length guess. A popular strategy is to assume that the first-order change in the function at iterate x_k will be the same as that obtained at the previous step, $k-1$: $\alpha_{k,0}\nabla f_k^T p_k = \alpha_{k-1}\nabla f_{k-1}^T p_{k-1}$, so that

$$\alpha_{k,0} = \alpha_{k-1}\frac{\nabla f_{k-1}^T p_{k-1}}{\nabla f_k^T p_k}$$

Another way is to look at the ratio of the initial guess at the previous step, $\alpha_{k-1,0}$ to the actual step taken for that iterate, α_{k-1}. If the initial guess was much smaller than the actual step, we should take $\alpha_{k,0}$ substantially larger than $\alpha_{k-1,0}$ (remember: small α_k's are in some sense "more dangerous" than large ones); if it was much larger, we should decrease it somewhat; if it was about right, we may want to attempt the same or a slightly larger initial guess at k-th iterate.

The heuristics above are just examples of strategies. There can be and certainly are more available. There is no sufficient evidence to proclaim any particular one "correct" or "best".

Problem 5.
It seems very natural to stop the iterations as soon as the norm of the gradient is small, i.e., when $\|\nabla f(x_k)\| \le \epsilon$. After all, we *are* looking for a point where the gradient is zero! However if our function is almost flat near the minimum, this technique may not work. Consider, for example, the function $f(x) = \epsilon x^2/200$. Its minimizer is $x^* = 0$, while the condition $\|\nabla f(x_k)\| \le \epsilon$ would tell us to stop at $|x_k| \le 100$ - in particular, an initial guess $x_0 = 100$ would appear to be close enough to the minimum to be acceptable. Needless to say, in most practical cases, 100 can hardly be considered "close" to 0. It is in cases like this that scaling the norm of the gradient against the value of the function at that point, $\|\nabla f(x_k)\| \le \epsilon|f(x_k)|$, is useful. The combined criterion $\|\nabla f(x_k)\| \le \epsilon(1+|f(x_k)|)$ is better because it works equally well in both a "normal" situation and a "flat" situation, since iterations stop when the norm of the gradient is the *smallest* value, ϵ or $\epsilon|f(x_k)|$.

Problem 6.
The first of the three MATLAB generated plots shows the step lengths for the three different methods and for the two different initial guesses. Note that for this problem, the nonlinear conjugate gradient method is better than steepest decent, and Newton's method is significantly superior to them both. The second plot shows convergence rates. Convergence is linear for steepest descent and nonlinear conjugate gradient; it is quadratic for Newton's method. The third plot included is the surface plot of Rosenbrock's function. A look at it suggests why the second initial guess is more challenging than the first one, in that it takes more iterations to converge. The shortest path from (-1.2,1) to the minimum "crosses the hill" (i.e., ascends first and descends afterwards), so the corresponding search directions are not acceptable. The

algorithms require the iterates follow a longer path around the "hill" along the "valleys". This does not appear to be much of a factor for the steepest descent algorithm, though.

A few notes on the implementation. We used the

$$\alpha_{k,0} = \alpha_{k-1} \frac{\nabla f_{k-1}^T p_{k-1}}{\nabla f_k^T p_k}$$

heuristic in choosing the initial guess of step size in each iteration; backtrack and sufficient decrease was used to improve on this initial guess. Convergence properties matched theoretical predictions fairly well: we observed linear convergence with a rate 0.9984 (almost equal to theoretical) of the steepest descent algorithm; and quadratic convergence for Newton's method. For a lack of an exact theoretical measure of convergence rate for the nonlinear CGM, we compared our actual convergence rate to that of a linear CGM and found NCGM to converge slower, which does not appear surprising.