Abstract. In this work we present the conjugate gradient algorithms with special attention on their definition. 40 nonlinear conjugate gradient algorithms are presented. For each of them we present the formula for $\beta_k$ definition or the main ingredients for algorithm definition. The conjugate gradient algorithms can be classified in 6 groups: classical, hybrid, modified, scaled, parametrized and accelerated.

Conjugate gradient algorithms are characterized by strong local and global convergence properties and low memory requirements. The history of these methods begins with the research of Magnus Hestenes at the Institute for Numerical Analysis and with independent work of Eduard Stiefel at the Technische Hochschule Zürich.

In this survey we focus on conjugate gradient methods for solving the nonlinear unconstrained optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x),$$

where $f: \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, bounded from below. Starting from an initial guess, a nonlinear conjugate gradient algorithm generates a sequence of points $\{x_k\}$, according to the following recurrence formula:

$$x_{k+1} = x_k + \alpha_k d_k,$$

where $\alpha_k$ is the step length, usually obtained by Wolfe line search,

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \rho \alpha_k g_k^T d_k,$$

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k,$$
with $0 < \rho < 1/2 \leq \sigma < 1$, and the directions $d_k$ are computed as:

$$d_{k+1} = -g_{k+1} + \beta_k s_k, \quad d_0 = -g_0.$$ 

Here $\beta_k$ is a scalar known as the conjugate gradient parameter, $g_k = \nabla f(x_k)$ and $s_k = x_{k+1} - x_k$. In the following $y_k = g_{k+1} - g_k$. Different conjugate gradient algorithms correspond to different choices for the parameter $\beta_k$. Therefore, a crucial element in any conjugate gradient algorithm is the formula definition of $\beta_k$. Any conjugate gradient algorithm has a very simple general structure as it is illustrated below.

### The prototype of Conjugate Gradient Algorithm

**Step 1.** Select the initial starting point $x_0 \in \text{dom } f$ and compute: $f_0 = f(x_0)$ and $g_0 = \nabla f(x_0)$. Set $d_0 = -g_0$ and $k = 0$.

**Step 2.** Test a criterion for stopping the iterations. For example, if $\|g_k\| \leq \varepsilon$, then stop; otherwise continue with step 3.

**Step 3.** Using the Wolfe line search conditions:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \rho \alpha_k g_k^T d_k,$$

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k,$$

with $0 < \rho < 1/2 \leq \sigma < 1$, determine the steplength $\alpha_k$.

**Step 4.** Update the variables as: $x_{k+1} = x_k + \alpha_k d_k$. Compute $f_{k+1}$ and $g_{k+1}$. Compute $y_k = g_{k+1} - g_k$ and $s_k = x_{k+1} - x_k$.

**Step 5.** Determine $\beta_k$.

**Step 6.** Compute the search direction as: $d_{k+1} = -g_{k+1} + \beta_k s_k$.

**Step 7.** Restart criterion. If the restart criterion of Powell $\left\| g_{k+1}^T g_k \right\| > 0.2 \left\| g_{k+1} \right\|^2$ is satisfied, then set $d_{k+1} = -g_{k+1}$.

**Step 8.** Compute the initial guess $\alpha_k = \alpha_{k-1} \left\| d_{k-1} \right\| / \left\| d_k \right\|$, set $k = k + 1$ and continue with step 2.

---

1. **Hestenes - Stiefel (HS)**

$$\beta_k^{HS} = \frac{y_k^T g_{k+1}}{y_k^T s_k}.$$

2. **Fletcher - Reeves (FR)**

$$\beta_k^{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}.$$

3. **Polak – Ribiére - Polyak (PRP)**

$$\beta_k^{PRP} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}.$$

---

\[ \beta^\text{PRP}_k = \frac{y^T_k g_{k+1}}{g^T_k g_k}. \]

4. Polak – Ribiére - Polyak plus (PRP+)\(^4\)
\[ \beta^\text{PRP}_k = \max \left\{ 0, \frac{y^T_k g_{k+1}}{g^T_k g_k} \right\}. \]

5. Conjugate Descent – Fletcher (CD)\(^5\)
\[ \beta^\text{CD}_k = -\frac{g^T_{k+1} y_{k+1}}{g^T_k d_k}. \]

6. Liu - Storey (LS)\(^6\)
\[ \beta^\text{LS}_k = -\frac{y^T_k g_{k+1}}{g^T_k d_k}. \]

7. Dai – Yuan (DY)\(^7\)
\[ \beta^\text{DY}_k = \frac{g^T_{k+1} y_{k+1}}{y^T_k s_k}. \]

8. Dai – Liao (DL)\(^8\)
\[ \beta^\text{DL}_k = \frac{g^T_{k+1}(y_k-t s_k)}{y^T_k s_k}. \]

9. Dai – Liao plus (DL+)\(^9\)
\[ \beta^\text{DL+}_k = \max \left\{ 0, \frac{y^T_k g_{k+1}}{y^T_k s_k} \right\} - t \frac{s^T_k g_{k+1}}{y^T_k s_k} . \]

10. Andrei - Sufficient Descent Condition (CGSD)\(^10\)
\[ \beta^\text{CGSD}_k = \frac{g^T_{k+1} y_{k+1}}{y^T_k s_k} - \frac{(y^T_k g_{k+1})(s^T_k g_{k+1})}{(y^T_k s_k)^2}. \]

11. Hybrid Dai - Yuan (hDY)\(^11\)

\[ \beta_k^{hDY} = \max\left\{ c\beta_k^{DY}, \min\left\{ \beta_k^{HS}, \beta_k^{DY} \right\} \right\}, \]
\[ c = -(1 - \sigma)/(1 + \sigma). \]

12. Hybrid Dai – Yuan zero (hDYz)\(^{12}\)
\[ \beta_k^{hDYz} = \max\left\{ 0, \min\left\{ \beta_k^{HS}, \beta_k^{DY} \right\} \right\} \]

13. Gilbert – Nocedal (GN)\(^{13}\)
\[ \beta_k^{GN} = \max\left\{ -\beta_k^{FR}, \min\left\{ \beta_k^{PRP}, \beta_k^{FR} \right\} \right\} \]

14. Hu – Storey (HuS)\(^{14}\)
\[ \beta_k^{HuS} = \max\left\{ 0, \min\left\{ \beta_k^{PRP}, \beta_k^{FR} \right\} \right\} \]

15. Touati-Ahmed and Storey (TaS)\(^{15}\)
\[ \beta_k^{TaS} = \begin{cases} \beta_k^{PRP} & 0 \leq \beta_k^{PRP} \leq \beta_k^{FR}, \\ \beta_k^{FR} & \text{otherwise} \end{cases} \]

\[ \beta_k^{LS-CD} = \max\left\{ 0, \min\left\{ \beta_k^{LS}, \beta_k^{CD} \right\} \right\} \]

17. Birgin – Martínez (BM)\(^{16}\)
\[ \beta_k^{BM} = \frac{(\theta_k y_k^T - s_k)^T g_{k+1}}{y_k^T s_k} , \]
where \( \theta_k \) is the spectral gradient:
\[ \theta_k = \frac{s_k^T y_k}{y_k^T s_k} . \]

18. Birgin – Martínez plus (BM+)
\[ \beta_k^{BM+} = \max\left\{ 0, \frac{\theta_k y_k^T g_{k+1}}{y_k^T s_k} - \frac{s_k^T g_{k+1}}{y_k^T s_k} \right\} . \]
where \( \theta_k \) is the spectral gradient:
\[ \theta_k = \frac{s_k^T y_k}{y_k^T s_k} . \]


19. Scaled Polak-Ribiére-Polyak (sPRP)

\[ \beta_k^{\text{sPRP}} = \frac{\theta_{k+1} y_k^T g_{k+1}}{\alpha_k \theta_k g_k^T g_k}, \]

where \( \theta_k \) is the spectral gradient:

\[ \theta_k = \frac{s_k^T s_k}{y_k^T s_k}. \]

20. Scaled Fletcher – Reeves (sFR)

\[ \beta_k^{\text{sFR}} = \frac{\theta_{k+1} g_{k+1}^T g_{k+1}}{\alpha_k \theta_k g_k^T g_k}, \]

where \( \theta_k \) is the spectral gradient:

\[ \theta_k = \frac{s_k^T s_k}{y_k^T s_k}. \]

21. Scaled Hestenes – Stiefel (sHS)

\[ \beta_k^{\text{sHS}} = \frac{s_k^T \nabla^2 f(x_{k+1}) g_{k+1}^T}{s_k^T \nabla^2 f(x_{k+1}) s_k}, \]

22. Daniel (D)

\[ \beta_k^{\text{D}} = \frac{s_k^T \nabla^2 f(x_{k+1}) g_{k+1}^T}{s_k^T \nabla^2 f(x_{k+1}) s_k}. \]

23. Andrei – Sufficient Descent Condition from PRP (A-prp)

\[ \beta_k^{\text{A-prp}} = \frac{1}{y_k^T s_k} \left( y_k^T g_{k+1} - (y_k^T y_k)(s_k^T g_{k+1}) \right). \]

24. Andrei – Sufficient Descent Condition from DY (ACGA)

\[ \beta_k^{\text{ACGA}} = \frac{y_k^T g_{k+1}}{y_k^T s_k} - \frac{(y_k^T s_k)(s_k^T g_{k+1})}{(y_k^T s_k)^2}. \]

25. Andrei – Sufficient Descent Condition from DY zero (ACGA+)

\[ \beta_k^{\text{ACGA+}} = \max \left\{ 0, \frac{y_k^T g_{k+1}}{y_k^T s_k} \left( 1 - \frac{s_k^T g_{k+1}}{y_k^T s_k} \right) \right\}. \]

---


20 N. Andrei, Another nonlinear conjugate gradient algorithm with sufficient descent conditions for unconstrained optimization. ICI Technical Reports, November 22, 2006

26. Convex combination of PRP and DY from conjugacy condition (CCOMB - Andrei)

\[ \beta_k^{CCOMB} = (1 - \theta_k)\beta_k^{PRP} + \theta_k\beta_k^{DY}, \]

where

\[ \theta_k = \theta_k^{CCOMB} = \frac{(y_k^T g_{k+1})(y_k^T s_k) - (y_k^T g_k)(g_k^T g_k)}{(y_k^T g_{k+1})(y_k^T s_k) - \|g_k\|^2}. \]

If \( \theta_k^{CCOMB} \leq 0 \), then \( \beta_k = \beta_k^{PRP} \). If \( \theta_k^{CCOMB} \geq 1 \), then \( \beta_k = \beta_k^{DY} \).

27. Convex combination of PRP and DY from Newton direction (NDOMB - Andrei)

\[ \beta_k^{NDOMB} = (1 - \theta_k)\beta_k^{PRP} + \theta_k\beta_k^{DY}, \]

where

\[ \theta_k = \theta_k^{NDOMB} = \frac{(y_k^T g_{k+1} - s_k^T g_{k+1})\|g_k\|^2 - (g_k^T g_k)(y_k^T s_k)}{\|g_{k+1}\|^2\|g_k\|^2 - (g_{k+1}^T y_k)(y_k^T s_k)}. \]

If \( \theta_k^{NDOMB} \leq 0 \), then \( \beta_k^{NDOMB} = \beta_k^{PRP} \). If \( \theta_k^{NDOMB} \geq 1 \), then \( \beta_k = \beta_k^{DY} \).

28. Convex combination of HS and DY from Newton direction (HYBRID - Andrei)

\[ \beta_k^{HYBRID} = (1 - \theta_k)\beta_k^{HS} + \theta_k\beta_k^{DY}, \]

where

\[ \theta_k = -\frac{s_k^T g_{k+1}}{g_k^T g_{k+1}}. \]

If \( \theta_k \leq 0 \), then \( \beta_k^{HYBRID} = \beta_k^{HS} \). If \( \theta_k \geq 1 \), then \( \beta_k^{HYBRID} = \beta_k^{DY} \).

29. Convex combination of HS and DY from Newton direction with modified secant condition (HYBRIDM - Andrei)

\[ \beta_k^{HYBRIDM} = (1 - \theta_k)\beta_k^{HS} + \theta_k\beta_k^{DY}, \]

where

\[ \theta_k = \left( \frac{\delta \eta_k}{s_k^T s_k} - 1 \right) \frac{s_k^T g_{k+1} - \frac{y_k^T g_{k+1}}{y_k^T s_k} \delta \eta_k}{g_k^T g_{k+1} + \frac{y_k^T g_{k+1}}{y_k^T s_k} \delta \eta_k}. \]

---


\( \delta \) is a parameter. For \( \delta = 0 \) we get HYBRID. If \( \theta_k \leq 0 \), then \( \beta_k^{HYBRID} = \beta_k^{HY} \). If \( \theta_k \geq 1 \), then \( \beta_k^{HYBRID} = \beta_k^{DY} \).

30. Guaranteed descent with efficient line search (CG_DESCENT - Hager and Zhang)\(^{26}\)

\[
d_{k+1} = -g_{k+1} + \bar{\beta}_k^{HZ} d_k, \quad d_0 = -g_0,
\]

\[
\bar{\beta}_k^{HZ} = \max\{\beta_k^{HZ}, \eta_k\},
\]

\[
\eta_k = \frac{1}{\|d_k\|} \min\{\eta, \|g_k\|\}, \quad \eta = 0.01
\]

\[
\beta_k^{HZ} = \frac{1}{y_k^T d_k} \left( y_k - 2 \frac{\|y_k\|^2}{y_k^T d_k} d_k \right)^T g_{k+1}
\]

31. Yabe – Takano (YT)\(^{27}\)

\[
\beta_k^{YT} = \frac{g_{k+1}^T (z_k - t s_k)}{d_k^T z_k},
\]

where \( z_k = y_k + \frac{\delta}{{\xi}_k} u_k, \quad \xi_k = 6(f_k - f_{k+1}) + 3(g_k + g_{k+1})^T s_k, \quad \delta \geq 0 \) is a constant and \( u_k \in \mathbb{R}^n \) satisfies \( s_k^T u_k \neq 0 \); for example \( u_k = s_k \).

32. Yabe – Takano plus (YT+)\(^{28}\)

\[
\beta_k^{YT+} = \max\left\{0, \frac{g_{k+1}^T z_k}{d_k^T z_k} \right\} - t \frac{g_{k+1}^T s_k}{d_k^T z_k},
\]

where \( z_k = y_k + \frac{\delta}{{\xi}_k} u_k, \quad \xi_k = 6(f_k - f_{k+1}) + 3(g_k + g_{k+1})^T s_k, \quad \delta \geq 0 \) is a constant and \( u_k \in \mathbb{R}^n \) satisfies \( s_k^T u_k \neq 0 \); for example \( u_k = s_k \).

33. BFGS preconditioned (CONMIN – Shanno and Phua)\(^{29}\)


**Step 1. Initialization.** Select $x_0 \in \mathbb{R}^n$, and the parameters $0 < \rho < 1/2 \leq \sigma < 1$. Compute $f(x_0)$ and $g_0 = \nabla f(x_0)$. Set $d_0 = -g_0$ and $\alpha_0 = 1/\|g_0\|$. Set $k = 0$.

**Step 2. Compute the initial guess:**

$$x_{k+1} = x_k + \alpha_k d_k.$$ Compute $f(x_{k+1})$, $g_{k+1}$ and $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.

**Step 3. Update the variables:**

$$x_k = x_k + \alpha_k d_k,$$

$$f(x_k), g_k,$$

$$s_k = x_k - x_{k-1}, y_k = g_k - g_{k-1}.$$

**Step 4. Test a criterion for stopping the iterations.** For example, if $\|g_k\| \leq \varepsilon$, then stop; otherwise continue with step 5.

**Step 5. Test for restart.** If the iteration $k$ is a multiple of $n$, or $g_k^T g_k \geq 0.2 \|g_k\|^2$, then compute $d_{k+1}$ as:

$$d_{k+1} = -s_k - \gamma g_{k+1} - \left[ 1 - \gamma \frac{y_k^T s_k}{y_k^T y_k} \right] s_k - \gamma \frac{y_k^T g_{k+1}}{y_k^T y_k}s_k + \gamma \frac{y_k^T g_k}{y_k^T y_k}s_k,$$

Consider: $s_i = d_k$, $y_i = y_k$, $k = k + 1$ and continue with step 2. Otherwise, continue with step 6.

**Step 6. Compute:**

$$d_{k+1} = -s_k g_{k+1} + \frac{s_k^T g_{k+1}}{y_k^T s_k} y_k - \left[ 1 + \frac{y_k^T g_{k+1}}{y_k^T y_k} \right] s_k - \gamma \frac{y_k^T g_k}{y_k^T y_k}s_k,$$

where the vectors $\hat{H}_k g_{k+1}$ and $\hat{H}_k y_k$ are computed as:

$$\hat{H}_k g_{k+1} = \frac{y_k^T s_k}{y_k^T y_k} g_{k+1} - \frac{s_k^T g_{k+1}}{y_k^T y_k} y_k + \left[ 2 \frac{s_k^T g_{k+1}}{y_k^T y_k} y_k - \frac{y_k^T g_{k+1}}{y_k^T y_k} s_k \right] s_k,$$

$$\hat{H}_k y_k = \frac{y_k^T s_k}{y_k^T y_k} y_k - \frac{s_k^T y_k}{y_k^T y_k} y_k + \left[ 2 \frac{s_k^T y_k}{y_k^T y_k} y_k - \frac{y_k^T y_k}{y_k^T y_k} s_k \right] s_k.$$

**Step 7. Scale direction:**

$$d_{k+1} = \left[ 2(f(x_{k+1}) - f(x_k)) / d_{k+1}^T g_{k+1} \right] d_{k+1},$$

set $k = k + 1$ and continue with step 2.

---

**34. Scaled BFGS preconditioned (SCALCG - Andrei)**

**Step 1. Initialization.** Select $x_0 \in \mathbb{R}^n$, and the parameters $0 < \rho < 1/2 \leq \sigma < 1$. Compute $f(x_0)$ and $g_0 = \nabla f(x_0)$. Set $d_0 = -g_0$ and $\alpha_0 = 1/\|g_0\|$. Set $k = 0$. 

---


Step 2. Line search. Compute $\alpha_k$ satisfying the Wolfe conditions. Update the variables $x_{k+1} = x_k + \alpha_k d_k$. Compute $f(x_{k+1}), g_{k+1}$ and $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.

Step 3. Test for continuation of iterations. If this test is satisfied the iterations are stopped, else set $k = k + 1$.

Step 4. Scaling factor computation. Compute $\theta_{k+1} = \frac{s_k^T s_k}{y_k^T s_k}$.

Step 5. Restart direction. Compute the (restart) direction $d_k$ as:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \theta_{k+1} \left( \frac{g_{k+1}^T s_k}{y_k^T s_k} \right) y_k - \left[ 1 + \theta_{k+1} \frac{y_k^T y_k}{y_k^T s_k} \right] g_{k+1}^T s_k \left( \theta_{k+1} - \frac{g_{k+1}^T y_k}{y_k^T s_k} \right) s_k,$$

Step 6. Line search. Compute the initial guess: $\alpha_k = \alpha_{k-1} \left\| d_{k-1} \right\| / \left\| d_k \right\|$. Using this initialization compute $\alpha_k$ satisfying the Wolfe conditions. Update the variables $x_{k+1} = x_k + \alpha_k d_k$. Compute $f(x_{k+1}), g_{k+1}$ and $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.

Step 7. Store: $\theta = \theta_k$, $s = s_k$ and $y = y_k$.

Step 8. Test for continuation of iterations. If this test is satisfied the iterations are stopped, else set $k = k + 1$.

Step 9. Restart. If the Powell restart criterion: $\left\| g_{k+1}^T g_k \right\|^2 > 0.2 \left\| g_{k+1} \right\|^2$, is satisfied, then go to step 4 (a restart step); otherwise continue with step 10 (a standard step).

Step 10. Standard direction. Compute the direction $d_k$ as:

$$d_{k+1} = -v + \left( \frac{g_{k+1}^T s_k}{y_k^T s_k} \right) y_k - \left[ 1 + \frac{y_k^T w}{y_k^T s_k} \right] g_{k+1}^T s_k \left( \frac{y_k^T w}{y_k^T s_k} \right) s_k,$$

where $v$ and $w$ are computed as:

$$v = \theta g_{k+1} - \theta \left( \frac{g_{k+1}^T s}{y^T s} \right) y + \left[ 1 + \theta \frac{y^T y}{y^T s} \right] g_{k+1}^T s \left( \theta - \frac{g_{k+1}^T y}{y^T s} \right) s,$$

and

$$w = \theta y_k - \theta \left( \frac{y_k^T s}{y^T s} \right) y + \left[ 1 + \theta \frac{y_k^T y}{y^T s} \right] y_k s \left( \theta - \frac{y_k^T y}{y^T s} \right) s,$$

with saved values $\theta$, $s$ and $y$.

Step 11. Line search. Compute the initial guess: $\alpha_k = \alpha_{k-1} \left\| d_{k-1} \right\| / \left\| d_k \right\|$. Using this initialization compute $\alpha_k$ satisfying the Wolfe conditions. Update the variables $x_{k+1} = x_k + \alpha_k d_k$. Compute $f(x_{k+1}), g_{k+1}$ and $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.

Step 12. Test for continuation of iterations. If this test is satisfied the iterations are stopped, else set $k = k + 1$ and go to step 9.

Remark:

In Step 4, $\theta_{k+1}$ can be computed in an anticipative way as: $\theta_{k+1} = \frac{1}{y_k^T s_k}$, where

$$\gamma_{k+1} = \frac{2}{d_k^T d_k} \frac{1}{\alpha_k} \left[ f(x_{k+1}) - f(x_k) - \alpha_k g_k^T d_k \right].$$

Observe that $\gamma_{k+1} > 0$ for convex functions. If $f(x_{k+1}) - f(x_k) - \alpha_k g_k^T d_k < 0$, then the reduction $f(x_{k+1}) - f(x_k)$ in function value is smaller than $\alpha_k g_k^T d_k$. In these cases the idea is to reduce a little the step size $\alpha_k$ as $\alpha_k - \eta_k$, maintaining the other quantities at their
values in such a way so that $\gamma_{k+1}$ is positive. To get a value for $\eta_k$ let us select a real $\delta > 0$, ”small enough” but comparable with the value of the function, and have

$$\eta_k = \frac{1}{g_k^T d_k} \left[ f(x_k) - f(x_{k+1}) + \alpha_k g_k^T d_k + \delta \right],$$

with which a new value for $\gamma_{k+1}$ can be computed as:

$$\gamma_{k+1} = \frac{2}{d_k^T d_k (\alpha_k - \eta_k)^2} \left[ f(x_{k+1}) - f(x_k) - (\alpha_k - \eta_k) g_k^T d_k \right].$$

35. Accelerated scaled BFGS preconditioned (ASCLCG - Andrei)\[31\]

**Step 1. Initialization.** Select $x_0 \in \mathbb{R}^n$, and the parameters $0 < \sigma_1 \leq \sigma_2 < 1$. Compute $f(x_0)$ and $g_0 = \nabla f(x_0)$. Set $d_0 = -g_0$ and $\alpha_0 = 1 / \|g_0\|$. Set $k = 0$.

**Step 2. Line search.** Compute $\alpha_k$ satisfying the Wolfe conditions. Update the variables $x_{k+1} = x_k + \alpha_k d_k$. Compute $f(x_{k+1}), g_{k+1}$ and $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.

**Step 3. Test for continuation of iterations.** If this test is satisfied the iterations are stopped, else set $k = k + 1$.

**Step 4. Scaling factor computation.** Compute $\theta_{k+1} = \frac{y_k^T s_k}{y_k^T s_k}$.

**Step 5. Restart direction.** Compute the (restart) direction $d_{k+1}$ as:

$$d_{k+1} = -\theta_{k+1} g_{k+1} + \theta_{k+1} \left( \frac{g_{k+1}^T s_k}{y_k^T s_k} \right) y_k - \left[ 1 + \theta_{k+1} \frac{y_k^T y_k}{y_k^T s_k} \right] \frac{g_{k+1}^T s_k}{y_k^T s_k} - \theta_{k+1} \frac{g_{k+1}^T y_k}{y_k^T s_k} s_k.$$

**Step 6. Line search.** Compute the initial guess: $\alpha_k = \alpha_{k-1} \|d_{k-1}\|_2 / \|d_k\|_2$. Using this initialization compute $\alpha_k$ satisfying the Wolfe conditions. Update the variables $x_{k+1} = x_k + \alpha_k d_k$. Compute $f(x_{k+1})$ and $g_{k+1}$.

**Step 7. Acceleration scheme.** Compute $a_k = g_k^T d_k$ and $b_k = (g_k - g_{k+1})^T d_k$. If $b_k \neq 0$, then compute $\gamma_k = a_k / b_k$ and update the variables as: $x_{k+1} = x_k + \gamma_k \alpha_k d_k$. Compute $f(x_{k+1})$, $g_{k+1}$ and $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$. Otherwise (if $b_k = 0$), then compute $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.

**Step 8. Store: $\theta = \theta_k$, $s = s_k$ and $y = y_k$.**

**Step 9. Test for continuation of iterations.** If this test is satisfied the iterations are stopped, else set $k = k + 1$.

**Step 10. Restart.** If the Powell restart criterion: $\|g_{k+1}^T g_k\| \geq 0.2 \|g_{k+1}\|$, is satisfied, then go to step 4 (a restart step); otherwise continue with step 11 (a standard step).

**Step 11. Standard direction.** Compute the direction $d_k$ as:

$$d_{k+1} = -v + \left( \frac{g_{k+1}^T s_k}{y_k^T s_k} \right) w + \left( \frac{g_{k+1}^T w}{y_k^T s_k} \right) s_k - \left[ 1 + \frac{y_k^T w}{y_k^T s_k} \right] \frac{g_{k+1}^T s_k}{y_k^T s_k} - \frac{g_{k+1}^T y_k}{y_k^T s_k} s_k,$$

where $v$ and $w$ are computed as:

---

\[ v = \theta g_{k+1} - \theta \left( \begin{bmatrix} g^T_{k+1} s \\ y^T s \end{bmatrix} y + \left[ \left( 1 + \theta \frac{y^T y}{y^T s} \right) g^T_{k+1} s - \theta \frac{g^T_{k+1} y}{y^T s} \right] s, \right. \]

and

\[ w = \theta y_k - \theta \left( \begin{bmatrix} y^T s \\ y^T s \end{bmatrix} y + \left[ \left( 1 + \theta \frac{y^T y}{y^T s} \right) y^T s - \theta \frac{y^T y}{y^T s} \right] s, \right. \]

with saved values \( \theta, s \) and \( y \).

**Step 12. Line search.** Compute the initial guess: \( \alpha_k = \alpha_{k-1} \left\| d_{k-1} \right\|_2 / \left\| d_k \right\|_2 \). Using this initialization compute \( \alpha_k \) satisfying the Wolfe conditions. Update the variables \( x_{k+1} = x_k + \alpha_k d_k \). Compute \( f(x_{k+1}) \) and \( g_{k+1} \).

**Step 13. Acceleration scheme.** Compute \( a_k = g_k^T d_k \) and \( b_k = (g_k - g_{k+1})^T d_k \). If \( b_k \neq 0 \), then compute \( y_k = a_k / b_k \) and update the variables as: \( x_{k+1} = x_k + y_k \alpha_k d_k \). Compute \( f(x_{k+1}) \), \( g_{k+1} \) and \( s_k = x_{k+1} - x_k \), \( y_k = g_{k+1} - g_k \). Otherwise (if \( b_k = 0 \)), then compute \( s_k = x_{k+1} - x_k \), \( y_k = g_{k+1} - g_k \).

**Step 14. Test for continuation of iterations.** If this test is satisfied the iterations are stopped, else set \( k = k + 1 \) and go to step 10.

36. **Accelerated conjugate gradient algorithm from Newton direction with modified secant condition** (ACGMSEC - Andrei)

**Step 1. Initialization.** Select the initial starting point \( x_0 \in \text{dom} f \) and compute: \( f_0 = f(x_0) \) and \( g_0 = \nabla f(x_0) \). Set \( d_0 = -g_0 \) and \( k = 0 \). Select a value for parameters \( \varepsilon \) and \( \tau \).

**Step 2. Test a criterion for stopping the iterations.** For example, if \( \left\| g_k \right\|_\infty \leq \varepsilon \), then stop; otherwise continue with step 3.

**Step 3. Line search.** Using the Wolfe line search conditions determine the steplength \( \alpha_k \).

**Step 4. Compute.** \( z = x_k + \alpha_k d_k \), \( g_z = \nabla f(z) \) and \( y_k = g_k - g_z \).

**Step 5. Compute.** \( a_k = g_k^T d_k \), and \( b_k = y_k^T d_k \).

**Step 6. Acceleration scheme.** If \( b_k \neq 0 \), then compute \( y_k = a_k / b_k \) and update the variables as: \( x_{k+1} = x_k + y_k \alpha_k d_k \), otherwise update the variables as \( x_{k+1} = x_k + \alpha_k d_k \). Compute \( f_{k+1} \) and \( g_{k+1} \). Compute \( y_k = g_{k+1} - g_k \) and \( s_k = x_{k+1} - x_k \).

**Step 7. Set.** \( \delta = 0 \). If \( \left\| s_k \right\| \leq \tau \), then set \( \delta = 1 \).

**Step 8. Determine** \( \beta_k \) as:

\[ \beta_k = \max \left\{ \frac{y_k^T g_{k+1}}{y_k^T s_k + \delta \eta_k}, 0 \right\} \left( 1 - \frac{\delta \eta_k}{\eta_k} \right) \frac{s_k^T g_{k+1}}{y_k^T s_k + \delta \eta_k} \]

**Step 9. Direction computation.** Compute the search direction as: \( d_{k+1} = -g_{k+1} + \beta_k s_k \).

**Step 10. Restart criterion.** If the restart criterion of Powell \( \left\| g_{k+1}^T g_k \right\| > 0.2 \left\| g_{k+1} \right\|^2 \) is satisfied, then set \( d_{k+1} = -g_{k+1} \).

---


Step 11. Compute the initial guess $\alpha_k = \alpha_{k-1} \frac{\|d_{k-1}\|}{\|d_k\|}$, set $k = k + 1$ and continue with step 2.

37. Accelerated conjugate gradient algorithm from Newton direction with finite difference HESSIAN / vector product approximation (ACGHES - Andrei)

Step 1. Initialization. Select the initial starting point $x_0 \in \text{dom } f$ and compute: $f_0 = f(x_0)$ and $g_0 = \nabla f(x_0)$. Set $d_0 = -g_0$ and $k = 0$. Select a value for the parameter $\epsilon$.

Step 2. Test a criterion for stopping the iterations. For example, if $\|g_k\|_\infty \leq \epsilon$, then stop; otherwise continue with step 3.

Step 3. Line search. Using the Wolfe line search conditions determine the steplength $\alpha_k$.

Step 4. Compute: $z = x_k + \alpha_k d_k$, $g_z = \nabla f(z)$ and $y_k = g_k - g_z$.

Step 5. Compute: $a_k = g_k^T d_k$, and $b_k = y_k^T d_k$.

Step 6. Acceleration scheme. If $b_k \neq 0$, then compute $\gamma_k = a_k / b_k$ and update the variables as $x_{k+1} = x_k + \gamma_k \alpha_k d_k$, otherwise update the variables as $x_{k+1} = x_k + \alpha_k d_k$. Compute $f_{k+1}$ and $g_{k+1}$. Compute $s_k = x_{k+1} - x_k$.

Step 7. Hessian / vector product approximation. Compute $\delta = \frac{2\sqrt{\epsilon_m} (1 + \|x_{k+1}\|)}{\|s_k\|}$ and $y_k = (\nabla f(x_{k+1} + \delta s_k) - \nabla f(x_{k+1})) / \delta$.

Step 8. Compute $\beta_k = (y_k^T g_{k+1} - s_k^T g_{k+1}) / s_k^T y_k$.

Step 9. Direction computation. Compute the search direction as $d_{k+1} = -g_{k+1} + \beta_k s_k$.

Step 10. Restart criterion. If the restart criterion of Powell $\|g_{k+1}^T g_{k+1}\| > 0.2 \|g_{k+1}\|^2$ is satisfied, then set $d_{k+1} = -g_{k+1}$.

Step 11. Compute the initial guess $\alpha_k = \alpha_{k-1} \frac{\|d_{k-1}\|}{\|d_k\|}$, set $k = k + 1$ and continue with step 2.

Remark:
In step 7 the computation of $\delta$ is implemented as:

$$\delta = \max \left\{ \frac{\varphi}{\max \{10\varphi, \|s_k\|\}}, \frac{\varphi}{100} \right\}, \quad \varphi = 2\sqrt{\epsilon_m} (1 + \|x_{k+1}\|/\sqrt{n}) .$$

38. Parametrized CG with one parameter

$$\beta_k = \frac{\|g_{k+1}\|^2}{\lambda_k \|g_k\|^2 + (1 - \lambda_k) d_k^T y_k}, \quad \lambda_k \in [0, 1].$$

The FR algorithm corresponds to $\lambda_k = 1$. The DY algorithm correspond to $\lambda_k = 0$.

---


39. Parametrized CG with two parameters\textsuperscript{35}

$$\beta_k = \frac{\mu_k \|g_{k+1}\|^2 + (1 - \mu_k)g_{k+1}^T y_k}{\lambda_k \|g_k\|^2 + (1 - \lambda_k) d_k^T y_k}, \quad \lambda_k, \mu_k \in [0, 1].$$

This two parameter family includes the methods: FR, DY, PRP and HS in extreme cases.

40. Parametrized CG with three parameters\textsuperscript{36}

$$\beta_k = \frac{\mu_k \|g_{k+1}\|^2 + (1 - \mu_k)g_{k+1}^T y_k}{(1 - \lambda_k - \omega_k) \|g_k\|^2 + \lambda_k d_k^T y_k - \omega_k d_k^T g_k}, \quad \lambda_k, \mu_k \in [0, 1] \quad \text{and} \quad \omega_k \in [0, 1 - \lambda_k].$$

This three parameter family includes the six classical conjugate gradient algorithms, as well as the previous one-parameter and two-parameter families.

March 14, 2008
