Pipelined Conjugate Gradient Method の導出

東京大学 森田直樹

2018 年 10 月 2 日

1 共役勾配法

共役勾配法は，Krylov 部分空間による反復法として広く知られている．共役勾配法は，正定値性対称行列を係数行列にもつ連立一次方程式 $Ax = b$ を反復的に解くことの他に，式 (1) の最小化問題を解くと考えることもできる．

$$f(x) = \frac{1}{2}x^T Ax - b^T x \tag{1}$$

図 1 に，共役勾配法を示した．ここで，A は $n \times n$ 次の係数行列，b は n 次の右辺ベクトル，r は残差ベクトル，p は探索方向ベクトル，ε は収束判定値である．

共役勾配法の特徴は，1 反復あたりの演算量とメモリ使用量が少ないこと，解くべき行列の次元数 N に対して理論上 N 回反復すれば収束することである．共役勾配法の収束のしやすさは，最大固有値と最小固有値の比で表される条件数 (Comdition number) を指標にすることができる．

2 前処理つき共役勾配法

共役勾配法は，解くべき行列の条件数が増大すると収束に必要な反復回数も増大することが知られている．そのため，係数行列の条件数を低減させることを目的として，元の方程式 $Ax = b$ を解くかわりに，方程式 (2) を解くことを考える．

$$M^{-1}Ax = M^{-1}b \tag{2}$$

ここで，M は左前処理行列である．この方程式 (2) に共役勾配法を適用したものが，前処理つき共役勾配法である．図 2 に，前処理つき共役勾配法を示した．

前処理行列 M は，$M \simeq A$ であるほど前処理の効果が大きくなるが，M^{-1} を計算する必要があることから計算量は増大する．そのため，前処理生成時間と，前処理行列によって短縮された反復回数分の計算時間のトレードオフとなる．例えば，前処理行列 $M = A$ として M^{-1} を完全に解けば 1 回の反復で解を得られるが，直接法を用いて解を得ることと同等の演算量とメモリ使用量が必要になる．

$r_0 = b - Ax_0$, $p_0 = r_0$
for $i = 1, 2, 3, \ldots,$ **do**
  $\alpha_i = (r_{i-1}, r_{i-1})/(p_{i-1}, Ap_{i-1})$
  $x_i = x_{i-1} + \alpha_i p_{i-1}$
  $r_i = r_{i-1} - \alpha_i A p_{i-1}$
  if $(\|r_i\|_2/\|r_0\|_2 \leqq \varepsilon)$ **then stop**
  $\beta_i = (r_i, r_i)/(r_{i-1}, r_{i-1})$
  $p_i = r_i + \beta_i p_{i-1}$
end for

図 1　Conjugate gradient method

$r_0 = b - Ax_0$, $u_0 = M^{-1} r_0$, $p_0 = u_0$
for $i = 1, 2, 3, \ldots,$ **do**
  $s = Ap_{i-1}$
  $\alpha_i = (r_{i-1}, u_{i-1})/(s, p_{i-1})$
  $x_i = x_{i-1} + \alpha_i p_{i-1}$
  $r_i = r_{i-1} - \alpha_i s_{i-1}$
  if $(\|r_i\|_2/\|r_0\|_2 \leqq \varepsilon)$ **then stop**
  $u_i = M^{-1} r_i$
  $\beta_i = (r_i, u_i)/(r_{i-1}, u_{i-1})$
  $p_i = u_i + \beta_i p_{i-1}$
end for

図 2　Preconditioned conjugate gradient method

3 Preconditioned pipelined conjugate gradient method

Preconditioned pipelined conjugate gradient method (以後，PipeCG) は，全体通信に必要な同期による遅延を隠蔽するために，Ghysels and Vanroose によって提案された手法である [1]．図 3 に，PipeCG を示した．ここで，z，q，s，u，z，w，n，m は一時ベクトル，γ，δ は一時定数である．PipeCG は，一般的な共役勾配法と数学的に等価なことが証明されているが，一時ベクトルを多く扱うことによる丸め誤差の影響が大きい．

本手法は，新たに追加した一時ベクトルによって，図 3 の 3，4 行目と 5，6 行目に関わる計算をオーバーラッピングすることが可能である．この実現のため，MPI 3.0 準拠の非同期全体通信関数 MPI_Iallreduce を用いて計算を行う．

反復中に必要なベクトルの本数は一般的な共役勾配法で x，r，s，u の 4 本，PipeCG で x，r，s，u，z，q，n，m，w の 9 本である．そのため，メモリ使用量と一反復あたりの演算量は増加するが，数値計算では効率的なメモリアクセスが可能であるため，計算時間の増加は無視できる[*1]．

1: $r_0 = b - Ax_0$, $u_0 = M^{-1}r_0$, $w_0 = Au_0$
2: **for** $i = 0, 1, 2, \ldots,$ **do**
3: $\quad \gamma_i = (r_i, u_i)$
4: $\quad \delta = (w_i, u_i)$
5: $\quad m_i = M^{-1}w_i$
6: $\quad n_i = Am_i$
7: $\quad$ **if** $i > 0$ **then**
8: $\qquad \beta_i = \gamma_i/\gamma_{i-1}$; $\alpha_i = \gamma_i/(\delta - \beta_i\gamma_i/\alpha_{i-1})$
9: $\quad$ **else**
10: $\qquad \beta_0 = 0$; $\alpha_0 = \gamma_0/\delta$
11: $\quad$ **end if**
12: $\quad z_i = n_i + \beta_i z_{i-1}$
13: $\quad q_i = m_i + \beta_i q_{i-1}$
14: $\quad s_i = w_i + \beta_i s_{i-1}$
15: $\quad p_i = u_i + \beta_i p_{i-1}$
16: $\quad x_{i+1} = x_i + \alpha_i p_i$
17: $\quad r_{i+1} = r_i - \alpha_i s_i$
18: $\quad u_{i+1} = u_i - \alpha_i q_i$
19: $\quad w_{i+1} = w_i - \alpha_i z_i$
20: $\quad$ **if** $(\|r_i\|_2/\|r_0\|_2 \leqq \varepsilon)$ **then stop**
21: **end for**

図 3 Preconditioned pipelined conjugate gradient method

[*1] FrontISTR/HECMW による数値実験で実証した．

4 Pipelined Conjugate Gradient Method の導出

1: $r_0 = b - Ax_0$, $p_0 = M^{-1}r_0$
2: **for** $i = 0, 1, 2, \ldots,$ **do**
3: $\quad \alpha = (r_i, M^{-1}r_i)/(p_i, Ap_i)$
4: $\quad x_{i+1} = x_i + \alpha p_i$
5: $\quad r_{i+1} = r_i - \alpha A p_i$
6: $\quad$ **if** $(\|r_i\|_2/\|r_0\|_2 \leqq \varepsilon)$ **then stop**
7: $\quad \beta = (r_{i+1}, M^{-1}r_{i+1})/(r_i, M^{-1}r_i)$
8: $\quad p_{i+1} = M^{-1}r_{i+1} + \beta p_i$
9: **end for**

図 4 Preconditioned conjugate gradient method (step 1)

図 4 に，前処理付き共役勾配法を示す．ここで，一時ベクトル $s = Ap_i$，$u_i = M^{-1}r_i$ を導入し，3 行目，7 行目，8 行目を書き換え，図 5 を得る．

1: $r_0 = b - Ax_0$, $u_0 = M^{-1}r_0$, $p_0 = M^{-1}r_0$
2: **for** $i = 0, 1, 2, \ldots,$ **do**
3: $\quad s = Ap_i$
4: $\quad \alpha = (r_i, u_i)/(p_i, s)$
5: $\quad x_{i+1} = x_i + \alpha p_i$
6: $\quad r_{i+1} = r_i - \alpha s_i$
7: $\quad$ **if** $(\|r_i\|_2/\|r_0\|_2 \leqq \varepsilon)$ **then stop**
8: $\quad u_{i+1} = M^{-1}r_{i+1}$
9: $\quad \beta = (r_{i+1}, u_{i+1})/(r_i, u_i)$
10: $\quad p_{i+1} = u_{i+1} + \beta p_i$
11: **end for**

図 5 Preconditioned conjugate gradient method (step 2)

図 5 に，一時変数 $\gamma_i = (r_i, u_i)$，$\phi = (p_i, s)$ を導入することで，4 行目，9 行目を書き換え，図 6 を得る．

1: $r_0 = b - Ax_0$, $u_0 = M^{-1}r_0$,
$p_0 = M^{-1}r_0$, $\gamma_0 = (r_0, u_0)$
2: **for** $i = 0, 1, 2, \ldots,$ **do**
3: $\quad s = Ap_i$
4: $\quad \phi = (p_i, s)$
5: $\quad \alpha = \gamma_i / \phi$
6: $\quad x_{i+1} = x_i + \alpha p_i$
7: $\quad r_{i+1} = r_i - \alpha s_i$
8: $\quad$ **if** $(\|r_i\|_2 / \|r_0\|_2 \leqq \varepsilon)$ **then stop**
9: $\quad u_{i+1} = M^{-1} r_{i+1}$
10: $\quad \gamma_{i+1} = (r_{i+1}, u_{i+1})$
11: $\quad \beta = \gamma_{i+1} / \gamma_i$
12: $\quad p_{i+1} = u_{i+1} + \beta p_i$
13: **end for**

図 6　Preconditioned conjugate gradient method (step 3)

初期条件 $u_0 = M^{-1}r_0$, $p_0 = M^{-1}r_0$, $\gamma_0 = (r_0, u_0)$ を CG ループ中で計算するように，9 行目，10 行目，11 行目，12 行目を移動，ループを変形し，図 7 を得る．$i = 0$ のとき，$p_0 = u_0$ を考慮して，β を定める．

$r_0 = b - Ax_0$
for $i = 0, 1, 2, \ldots,$ **do**
$\quad u_i = M^{-1} r_i$
$\quad \gamma_i = (r_i, u_i)$
$\quad$ **if** $i > 0$ **then**
$\qquad \beta = \gamma_i / \gamma_{i-1}$
$\quad$ **else**
$\qquad \beta = 0$
$\quad$ **end if**
$\quad p_i = u_i + \beta p_{i-1}$
$\quad s = Ap_i$
$\quad \phi = (p_i, s)$
$\quad \alpha = \gamma_i / \phi$
$\quad x_{i+1} = x_i + \alpha p_i$
$\quad r_{i+1} = r_i - \alpha s_i$
$\quad$ **if** $(\|r_i\|_2 / \|r_0\|_2 \leqq \varepsilon)$ **then stop**
end for

図 7　Preconditioned conjugate gradient method (step 4)

ここで，一時ベクトル $w_i = Au_i$ を導入し，$p_i = u_i + \beta p_{i-1}$ の関係から，$s = Ap_i$ を式 (3) のように変形する．

$$
\begin{aligned}
s_i &= Ap_i \\
&= A(u_i + \beta p_{i-1}) \\
&= Au_i + \beta A p_{i-1} \\
&= w_i + \beta s_{i-1}
\end{aligned} \tag{3}
$$

さらに，一時変数 $\delta = (u_i, w_i)$ を導入し，$\alpha = (r_i, u_i)/(p_i, s_i)$，$\gamma_i = (r_i, u_i)$，$r_{i+1} = r_i - \alpha s_i$ の関係から，$\phi = (p_i, s)$ を式 (2) のように変形する．このとき，$(p_{i-1}, Ap_i) = 0$ の共役性を考慮する．$(M^{-1}r_i, r_{i-1})$ に関する関係は，式 (4) を示す．

$$
\begin{aligned}
\phi &= (p_i, s_i) \\
&= (u_i + \beta p_{i-1}, w_i + \beta s_{i-1}) \\
&= (u_i, w_i) + \beta(u_i, s_{i-1}) + \beta(p_{i-1}, w_i) + \beta^2(p_{i-1}, s_{i-1}) \\
&= (u_i, w_i) + \beta(u_i, s_{i-1}) + \beta(p_{i-1}, w_i + \beta s_{i-1}) \\
&= (u_i, w_i) + \beta(u_i, s_{i-1}) + \beta(p_{i-1}, s_i) \\
&= (u_i, w_i) + \beta(u_i, s_{i-1}) + \beta(p_{i-1}, Ap_i) \\
&= (u_i, w_i) + \beta(u_i, s_{i-1}) + 0 \\
&= (u_i, w_i) + \beta\frac{\alpha_{i-1}}{\alpha_{i-1}}(u_i, s_{i-1}) \\
&= (u_i, w_i) + \frac{\beta}{\alpha_{i-1}}(u_i, \alpha_{i-1}s_{i-1}) \\
&= (u_i, w_i) + \frac{\beta}{\alpha_{i-1}}(u_i, -r_i + r_{i-1}) \\
&= (u_i, w_i) - \frac{\beta}{\alpha_{i-1}}(u_i, r_i) + \frac{\beta}{\alpha_{i-1}}(u_i, r_{i-1}) \\
&= \delta - \frac{\beta}{\alpha_{i-1}}\gamma_i + \frac{\beta}{\alpha_{i-1}}(M^{-1}r_i, r_{i-1}) \\
&= \delta - \frac{\beta}{\alpha_{i-1}}\gamma_i + 0 \\
&= \delta - \frac{\beta}{\alpha_{i-1}}\gamma_i
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
(M^{-1}r_i, r_{i-1}) &= ((LL^T)^{-1}r_i, r_{i-1}) \\
&= (L^{-T}L^{-1}r_i, r_{i-1}) \\
&= r_{i-1}^T L^{-T} L^{-1} r_i \\
&= (L^{-1}r_{i-1})^T L^{-1} r_i \\
&= \tilde{r}_{i-1}^T \tilde{r}_i \\
&= (\tilde{r}_i, \tilde{r}_{i-1}) \\
&= 0
\end{aligned} \tag{5}
$$

一次ベクトル $w_i = Au_i$，一次変数 $\delta = (u_i, w_i)$ と，式 (4)，(5) より得た $\phi = \delta - \frac{\beta}{\alpha_{i-1}}\gamma_i$ から，図 7 の 12 行目，13 行目の α に関する計算を書き直し，図 8 を得る．

1: $r_0 = b - Ax_0$
2: **for** $i = 0, 1, 2, \ldots,$ **do**
3: $\quad u_i = M^{-1} r_i$
4: $\quad w_i = Au_i$
5: $\quad \gamma_i = (r_i, u_i)$
6: $\quad$ **if** $i > 0$ **then**
7: $\qquad \beta = \gamma_i / \gamma_{i-1}$
8: $\quad$ **else**
9: $\qquad \beta = 0$
10: $\quad$ **end if**
11: $\quad p_i = u_i + \beta p_{i-1}$
12: $\quad s = Ap_i$
13: $\quad \delta = (u_i, w_i)$
14: $\quad \alpha_i = \gamma_i / (\delta - \beta \gamma_i / \alpha_{i-1})$
15: $\quad x_{i+1} = x_i + \alpha_i p_i$
16: $\quad r_{i+1} = r_i - \alpha_i s_i$
17: $\quad$ **if** $(\|r_i\|_2 / \|r_0\|_2 \leqq \varepsilon)$ **then stop**
18: **end for**

図 8　Preconditioned conjugate gradient method (step 5)

図 8 の 13 行目の δ の計算は，2 行目，3 行目の計算を終えた時点で計算可能である．さらに，14 行目の α の計算は，δ と β が更新された後に計算可能である．以上を書き直して，図 9 を得る．

1: $r_0 = b - Ax_0$
2: **for** $i = 0, 1, 2, \ldots,$ **do**
3: $\quad u_i = M^{-1} r_i$
4: $\quad w_i = Au_i$
5: $\quad \gamma_i = (r_i, u_i)$
6: $\quad \delta = (u_i, w_i)$
7: $\quad$ **if** $i > 0$ **then**
8: $\qquad \beta = \gamma_i / \gamma_{i-1};\ \alpha_i = \gamma_i / (\delta - \beta \gamma_i / \alpha_{i-1})$
9: $\quad$ **else**
10: $\qquad \beta = 0;\ \alpha_i = \gamma_i / \delta$
11: $\quad$ **end if**
12: $\quad p_i = u_i + \beta p_{i-1}$
13: $\quad s = Ap_i$
14: $\quad x_{i+1} = x_i + \alpha_i p_i$
15: $\quad r_{i+1} = r_i - \alpha_i s_i$
16: $\quad$ **if** $(\|r_i\|_2 / \|r_0\|_2 \leqq \varepsilon)$ **then stop**
17: **end for**

図 9　Preconditioned conjugate gradient method (step 6)

図 9 から，初期条件 $u_0 = M^{-1}r_0$，$w_0 = Au_0$ をあらかじめ計算することで，ループ内の計算順序を変更する．

$r_0 = b - Ax_0$, $u_0 = M^{-1}r_0$, $w_0 = Au_0$
for $i = 0, 1, 2, \ldots,$ **do**
  $\gamma_i = (r_i, u_i)$
  $\delta = (u_i, w_i)$
  if $i > 0$ **then**
    $\beta = \gamma_i/\gamma_{i-1}$; $\alpha_i = \gamma_i/(\delta - \beta\gamma_i/\alpha_{i-1})$
  else
    $\beta = 0$; $\alpha_i = \gamma_i/\delta$
  end if
  $p_i = u_i + \beta p_{i-1}$
  $s = Ap_i$
  $x_{i+1} = x_i + \alpha_i p_i$
  $r_{i+1} = r_i - \alpha_i s_i$
  $u_{i+1} = M^{-1}r_{i+1}$
  $w_{i+1} = Au_{i+1}$
  if $(\|r_i\|_2/\|r_0\|_2 \leqq \varepsilon)$ **then stop**
end for

図 10　Preconditioned conjugate gradient method (step 7)

式 (3) と一次ベクトル $q_i = M^{-1}s_i$，$z_i = Aq_i$ を導入し，以下の式 (6), (7), (8), (9) を得る．

$$\begin{aligned} u_{i+1} &= M^{-1}r_{i+1} \\ &= M^{-1}(r_i - \alpha_i s_i) \\ &= M^{-1}r_i - \alpha_i M^{-1}s_i \\ &= u_i - \alpha_i q_i \end{aligned} \tag{6}$$

$$\begin{aligned} w_{i+1} &= Au_{i+1} \\ &= A(u_i - \alpha_i q_i) \\ &= Au_i - \alpha_i Aq_i \\ &= w_i - \alpha_i z_i \end{aligned} \tag{7}$$

$$\begin{aligned} q_i &= M^{-1}s_i \\ &= M^{-1}(w_i + \beta s_{i-1}) \\ &= M^{-1}w_i + \beta M^{-1}s_{i-1} \\ &= M^{-1}w_i + \beta q_{i-1} \end{aligned} \tag{8}$$

$$\begin{aligned} z_i &= Aq_i \\ &= A(M^{-1}w_i + \beta q_{i-1}) \\ &= AM^{-1}w_i + \beta Aq_{i-1} \\ &= AM^{-1}w_i + \beta z_{i-1} \end{aligned} \tag{9}$$

これらの関係を図 7 に導入し，図 11 を得る．

1: $r_0 = b - Ax_0$, $u_0 = M^{-1}r_0$, $w_0 = Au_0$
2: **for** $i = 0, 1, 2, \ldots,$ **do**
3: $\quad \gamma_i = (r_i, u_i)$
4: $\quad \delta = (u_i, w_i)$
5: $\quad$ **if** $i > 0$ **then**
6: $\qquad \beta = \gamma_i/\gamma_{i-1}$; $\alpha_i = \gamma_i/(\delta - \beta\gamma_i/\alpha_{i-1})$
7: $\quad$ **else**
8: $\qquad \beta = 0$; $\alpha_i = \gamma_i/\delta$
9: $\quad$ **end if**
10: $\quad p_i = u_i + \beta p_{i-1}$
11: $\quad s_i = w_i + \beta s_{i-1}$
12: $\quad q_i = M^{-1}w_i + \beta q_{i-1}$
13: $\quad z_i = AM^{-1}w_i + \beta z_{i-1}$
14: $\quad x_{i+1} = x_i + \alpha_i p_i$
15: $\quad r_{i+1} = r_i - \alpha_i s_i$
16: $\quad u_{i+1} = u_i - \alpha_i q_i$
17: $\quad w_{i+1} = w_i - \alpha_i z_i$
18: $\quad$ **if** $(\|r_i\|_2/\|r_0\|_2 \leqq \varepsilon)$ **then stop**
19: **end for**

図 11　Preconditioned conjugate gradient method (step 8)

さらに，一次ベクトル $m_i = M^{-1}w_i$，$n_i = Am_i$ を導入すると，図 11 の 3〜11 行目とは独立に計算可能であることがわかる．よって，3 行目，4 行目の内積計算と独立して計算可能なように計算順序を変更し，図 12 の Preconditioned pipelined conjugate gradient method が導出される．

$r_0 = b - Ax_0$, $u_0 = M^{-1}r_0$, $w_0 = Au_0$
for $i = 0, 1, 2, \ldots,$ **do**
  $\gamma_i = (r_i, u_i)$
  $\delta = (u_i, w_i)$
  $m_i = M^{-1}w_i$
  $n_i = Am_i$
  if $i > 0$ **then**
    $\beta = \gamma_i/\gamma_{i-1}$; $\alpha_i = \gamma_i/(\delta - \beta_i\gamma_i/\alpha_{i-1})$
  else
    $\beta = 0$; $\alpha = \gamma_0/\delta$
  end if
  $p_i = u_i + \beta_i p_{i-1}$
  $s_i = w_i + \beta_i s_{i-1}$
  $q_i = m_i + \beta_i q_{i-1}$
  $z_i = n_i + \beta_i z_{i-1}$
  $x_{i+1} = x_i + \alpha_i p_i$
  $r_{i+1} = r_i - \alpha_i s_i$
  $u_{i+1} = u_i - \alpha_i q_i$
  $w_{i+1} = w_i - \alpha_i z_i$
  if $(\|r_i\|_2/\|r_0\|_2 \leqq \varepsilon)$ **then stop**
end for

図 12　Preconditioned pipelined conjugate gradient method

FrontISTR に導入する段階では，残差 $\|r_i\|_2$ の内積計算が必要なこと，通信オーバーヘッドの増大を避けるため集団通信の呼び出しを 1 度で済ませることを考慮して，図 13 のような実装を行った*2．通信隠蔽が行われる個所は，6 行目の非同期集団通信を call し 9 行目のバッファ確認を行う間の，7 行目，8 行目の行列ベクトル積を計算する部分である．

この手法の前処理として局所前処理を採用した場合，陽に通信が発生する箇所は図 10 の 8 行目，ベクトル m_i の袖領域に関する通信のみである．袖領域の通信は隣接する領域のみで完結するので，原著論文 (Ghysels 2014) では集団通信に対して通信時間を無視できるとしている．この点については，詳細な計算時間を測定し検討する．

*2 PETSc による実装では，集団通信を個別に呼んでいる

$r_0 = b - Ax_0$, $u_0 = M^{-1}r_0$, $w_0 = Au_0$

for $i = 0, 1, 2, \ldots,$ **do**

  $\gamma = (r_i, u_i)$ on local process

  $\delta = (u_i, w_i)$ on local process

  $\|r_i\|_2 = (r_i, r_i)$ on local process

  MPI_Iallreduce(γ, δ, $\|r_i\|_2$) with MPI_SUM

  $m_i = M^{-1}w_i$

  $n_i = Am_i$

  MPI_wait()

  if $(\|r_i\|_2 / \|r_0\|_2 \leqq \varepsilon)$ **then stop**

  if $i > 0$ **then**

    $\beta = \gamma/\gamma_{old}$; $\alpha = \gamma/(\delta - \beta\gamma/\alpha_{old})$

  else

    $\beta = 0$; $\alpha = \gamma/\delta$

  end if

  $p_i = u_i + \beta p_{i-1}$

  $s_i = w_i + \beta s_{i-1}$

  $q_i = m_i + \beta q_{i-1}$

  $z_i = n_i + \beta z_{i-1}$

  $x_{i+1} = x_i + \alpha p_i$

  $r_{i+1} = r_i - \alpha s_i$

  $u_{i+1} = u_i - \alpha q_i$

  $w_{i+1} = w_i - \alpha z_i$

  $\alpha_{old} = \alpha$

  $\gamma_{old} = \gamma$

end for

図 13 Preconditioned pipelined conjugate gradient method (FrontISTR implementation)

参考文献

[1] P. Ghysels and W. Vanroose, Hiding Global Synchronization Latency in the Preconditioned Conjugate Gradient Method, Parallel Computing, Vol.40 pp.224-238 (2014), Elsevier.