The main results bound $e^x = A^{-1}$.

Nigrous literature and folklore.

Arithmetic is fairly complicated and is also the subject of much

Convergence of CG in exact arithmetic. Its behavior under finite

There are several useful results in the literature concerning the

Convergence of CG
\[
\frac{1 + 2y^2}{1 - 2y^2} = a
\]

\[
V\|v_0\| \geq 2a_F \quad \forall \theta \in \mathbb{R}^n
\]

By the initial error \( v_0 \) and the number of \( A \), the error \( \epsilon_i \) is bounded in terms of \( \theta \) and the condition number of \( A \).
\[
\frac{1 + \sqrt{y}}{1 - \sqrt{y}} = \alpha
\]

\[
\|e^0\|_2 \geq 2\sqrt{\text{trace}(\beta_0^2 e^0)}
\]

**Theorem:** For $\mathbf{C}$, $\mathbf{E}$, is bounded in terms of $\alpha$, the condition number of $\mathbf{A}$ and the initial error $e^0$, by...
positive definite we have $\lambda > 0$, $\rho \equiv 1$, \\
whose columns are the eigenvectors of $A$. If $A$ is symmetric, $A$ \\
is always diagonalizable via the orthogonal matrix $V$.

$$
\begin{align*}
(\begin{pmatrix} \lambda_1 \\
\lambda_2 \\
\vdots \\
\lambda_n \end{pmatrix})_{\text{diag}} & = V \\
I & = DD \, D \\
LDDL & = V \\
VDD & = DDV
\end{align*}
$$

For a symmetric matrix, $\rho$ is the ratio of the largest to smallest

eigenvalue of the matrix.
THEOREM: If $A$ has $m$ distinct eigenvalues, i.e., there are $m$ eigenvalues, they play an important role in characterizing the convergence of CG.

For $I > I'$, then CG converges in at most $r$ steps.

$$u \geq r = \text{rank}(B)$$

where $B + I = A$. Theorem: If $A$ has $m$ distinct eigenvalues, i.e., there are $m$ eigenvalues.
almost converged after \( t \) steps.

If \( A \) is close to a rank \( r \) update to the identity then \( CG \) is

- "looks" like there are fewer distinct values.

- The spread of eigenvalues is getting small and therefore it

- \( CG \) converges quickly in the \( A \)-norm if \( r \approx 1 \). (This implies

Heuristically we have the following statements:
the resulting improvement in performance.

Transforming the system – or \textbf{preconditioning} the system – and transforming the system naturally, there is a tradeoff in the cost of coefficients. Ideally, the matrix can be transformed so that the eigenvalues, ideally, the matrix to have, effectively, fewer distinct eigenvalues. Ideally, the matrix can be transformed so that the system in order to improve the convergence rate. This is done by \textbf{preconditioning} in practice. In most cases it is necessary to alter the convergence in practice. Typically they do not yield satisfactory theorems are nice they typically do not yield satisfactory.

While finite termination and the distinct eigenvalue convergence

\textbf{Preconditioning}
can be written in terms of a simple modification of the earlier form.

After some tedious manipulation GC applied to the system above

\[ q_{t-1} \mathcal{C} = x \mathcal{C} + A \mathcal{C}_{t-1} \mathcal{C} \]

Apply GC to

\[ h = n \mathcal{W} \mathcal{C} = W \mathcal{C} \]

a matrix \( \mathcal{W} \) which is symmetric positive definite,

Find a matrix \( \mathcal{C} \) which is
end

\[ \gamma d \gamma G + 1 + \gamma z = 1 + \gamma d \]
\[ \gamma z \frac{\gamma l}{1 + \gamma l} + \gamma x = \gamma G \]
\[ 1 + \gamma l = 1 + \gamma z M \]

solve

\[ \gamma o \gamma o - \gamma l = 1 + \gamma l \]
\[ \gamma d \gamma o + \gamma x = 1 + \gamma x \]
\[ \gamma o \frac{\gamma d}{\gamma z} \frac{\gamma l}{l} = \gamma o \]
\[ \gamma d v = \gamma o \]

\[ \cdots \]

\[ 0 \gamma = \gamma, I, 0 \gamma = 0 \gamma M \]

solve

\[ 0 x = 0 d, 0 l = 0 x M \]

arbitrary; \[ 0 x \]

Preconditioned Conjugate Gradient
• Each iteration requires the solution of the preconditioning equation.

• This must be kept low in operations (or highly parallel) in order to maintain simplicity of each iteration which tends to contradict the requirement that \( M \) approximates \( A \) well.
Highly parallel and low in cost in terms of operations. In some cases, this can significantly improve convergence. It is often used.

Diagonal Preconditioning

Choose $M$ to be the diagonal elements of $A$. By definition, $M$ is positive definite. In practice, the block diagonal portion of $A$ is often used. (In practice, the block diagonal portion of $A$ is often used.)
Eigenvectors of $C_{-1}AV_{-1}$ are (approximately) $(1,0.01,0.99,1.00,1.03,0.97)$.

Eigenvectors of $A$ are (approximately) $(1000,100.01,9.99,1.00,1.0)$.

\[
\begin{pmatrix}
1 & -0.032 & 0 \\
-0.032 & 1 & -0.003 \\
0 & -0.003 & 1 \\
0 & 1 & -0.032 \\
1 & -0.003 & 1
\end{pmatrix} = \begin{pmatrix}
1 & -0.032 & 0 \\
-0.032 & 1 & -0.003 \\
0 & -0.003 & 1 \\
0 & 1 & -0.032 \\
1 & -0.003 & 1
\end{pmatrix} = A
\]

Example:
Incomplete Cholesky

\( (\forall) d \approx \forall_{-1} \forall_{I} \) is minimized.

Polynomial preconditioning:

Approximate inverse: \( d_1 W - I \) is minimized.

Other preconditioners:

Unfortunately, diagonal (or even block diagonal) preconditioning is not good enough.
As a result, \( T \) is as sparse as the lower part of \( A \) and \( M \) is used in PCG.

The matrix \( A \) and only updating elements in the active part that were nonzero in and only updating elements in the active part that were nonzero in computing by ignoring all-in during the Cholesky factorization is costly in terms of time and space the incomplete Cholesky factor \( L \) rather then computing the Cholesky factor of \( A \) which could be incomplete Cholesky.
end for
end for
end for
end if
\[(\forall,\exists)^V/!(\forall,\exists)^V \setminus (\exists,\forall)^V = (\exists,\forall)^V\]
\[\text{then } (0 \neq (\exists,\forall)^V)\]
\[u,\exists i = \exists\]
\[u, \exists i + \forall = \exists\]
\[\text{for}\]
\[(\forall,\forall)^V/!(\forall,\exists)^V = (\forall,\exists)^V\]
\[\text{then } (0 \neq (\forall,\exists)^V)\]
\[u, \forall i = \forall\]
\[\text{for}\]
so that a small number of iterations can be expected.

availability of a good preconditioner due to application attributes

that takes into account the structure of the matrix or the

iterative methods typically imply the lack of a fast direct method

that are Toeplitz or have low displacement rank. The use of

structured dense systems such as those with coefficient matrices

iterative methods are also becoming increasingly important for
The matrix is also called circulant since the rows wrap around.

\[
\begin{pmatrix}
  a_1 & a_4 & a_2 & a_3 \\
  a_2 & a_1 & a_4 & a_3 \\
  a_3 & a_2 & a_1 & a_4 \\
  a_4 & a_3 & a_2 & a_1 \\
\end{pmatrix}
\]

Consider the following Toeplitz matrix.
have reasonable parallel versions. can both be computed using FPL in $O(n \log n)$ operations and

\[
q_{-1} = x \\
q = z
\]

Therefore, computation of the Fourier transform of a vector and $V$ is a diagonal matrix. Moreover, where $V$ is a unitary matrix whose application is equivalent to the

\[
HV*H = C
\]

For any circulant matrix $C$, we have
Now consider the following Toeplitz matrix and its associated circulant matrix.
\[
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{pmatrix}
= \mathcal{O}
\]
circulant matrix-vector multiplication.

So $x_L$ can be found by using FFT's of approximately length 2n via

$$
\begin{pmatrix}
x_B \\
x_L
\end{pmatrix} =
\begin{pmatrix}
0 \\
x
\end{pmatrix}
\begin{pmatrix}
\mathcal{F} & B \\
A & L
\end{pmatrix} = \psi_C
$$
\[ d \| L_{1} - C - I \| \]

(Tyrishnikov, 1988)

or an approximation to the inverse of \( L \) can be generated so that

\[ d \| L - C \| \]

Determine a circulant matrix \( C \) such that

Toeplitz.

Suppose \( L \) is symmetric positive definite and a Toeplitz matrix. Suppose \( L \) is symmetric positive definite and

Circulant matrices can also be used to generate a preconditioner.
parallel implementation.

PCG with \( O(n \log n) \) operations per iteration and have a natural
matrix vector product for a circulant matrix we can implement
Given the Toeplitz matrix vector product and the fast solution of

where $\mathcal{L}$ is the transform domain version of $L$.

\[
\begin{align*}
\mathcal{A} \| \mathcal{L} - V \| &= \\
\mathcal{A} \| \mathcal{A} \mathcal{L} \mathcal{A} - V \| &= \\
\mathcal{A} \| \mathcal{A} \mathcal{L} \mathcal{A} - \mathcal{A} \mathcal{C} \mathcal{A} \| &= \\
\mathcal{A} \| \mathcal{A} (\mathcal{L} - \mathcal{C}) \mathcal{A} \| &= \mathcal{A} \| \mathcal{L} - \mathcal{C} \|
\end{align*}
\]

This is unitarily invariant so we have $\mathcal{A} \| x \|$. We have already seen that the first preconditioner can be related to a preconditioner that we
precconditioning.

domain and create $\mathcal{L}$ directly then use $G$ with diagonal
Some applications, e.g., adaptive filtering, work in the transform
precconditioning in the transform domain.
So Tony Chan’s preconditioner is equivalent to diagonal

$$(\mathcal{L})_{\text{diag}} p = v$$

Clearly, the error is minimized by