

Contents

1	Foundations	3
1.1	Projectors	3
1.1.1	Orthogonal Projector	3
1.1.2	Oblique Projector	4
1.1.3	Projector Geometric Intuitions	5
1.1.4	Projector as a 2-Norm Minimizer	5
1.2	Subspace Projection Methods	5
1.2.1	Prototype Algorithm	6
1.2.2	Energy Norm Minimization using Gradient	6
1.3	Krylov Subspace	8
1.3.1	The Grade of a Krylov Subspace	8
1.4	Useful Theorems and Mathematical Entities	10
1.4.1	Minimal Polynomial of a Matrix	10
1.4.2	Cauchy Interlace Theorem	10
1.4.3	Caley Hamilton's Theorem	10
1.4.4	The Chebyshev Polynomial	10
1.5	Deriving Conjugate Gradient from First Principles	10
1.5.1	CG Objective and Framework	10
1.5.2	Using the Projector	11
1.5.3	Assisted Conjugate Gradient	12
1.5.4	Properties of Assisted Conjugate Gradient	13
1.5.5	Residual Assisted Conjugate Gradient	14
1.5.6	RACG and Krylov Subspace	16
1.6	Arnoldi Iterations and Lanczos	17
1.6.1	The Arnoldi Iterations	17
1.6.2	Arnoldi Produces Orthogonal Basis for Krylov Subspace	19
1.6.3	The Lanczos Iterations	19
2	Analysis of Conjugate Gradient and Lanczos Iterations	21
2.1	Terminations Conditions of RACG	21
2.2	Terminations Conditions of The Lanczos Iterations	21
2.3	Convergence Rate of RACG under Exact Arithematic	21
2.4	From Conjugate Gradient to Lanczos	21
2.5	From Lanczos to Conjugate Gradient	21
3	Effects of Floating Point Arithematic	21
4	Appendix	21
5	Biliography	21

Notations

1. $\text{ran}(A) := \{Ax : \forall x \in \mathbb{R}^n\}$, $A \in \mathbb{R}^{m \times n}$, The range of a matrix.
2. $(A)_{i,j}$: The element in i th row and j th column of the matrix A .
3. $(A)_{i:j,i':j'}$: The submatrix whose top left corner is the (i, j) element in matrix A , and whose' right bottom corner is the (i', j') element in the matrix A . The notation is similar to matlab's rules for indexing.

Introduction

1 Foundations

This sections focuses on important mathematical entities that are important for formulating, analyzing the Conjugate Gradient and the Lanczos Algorithm. Major parts of this sections cited from... (citations here)

1.1 Projectors

A projector is a type of idempotent matrix. Its properties have importance for subspace projection method. In this section, we go through 2 types of projectors, the orthogonal projector and the oblique projector. The oblique projector is made useful for the derivation of the classic CG algorithm, which is referred to as RACG in the context of this paper. The orthogonal projector is made useful for the interpretations of Arnoldi Iterations.

Definition 1. A matrix P is a projector when:

$$P^2 = P \quad (1.1.1)$$

This property is sometimes referred as *idempotent*. As a consequence, $\text{ran}(I - P) = \text{null}(P)$ and here is the proof:

Proof.

$$\forall x \in \mathbb{C}^n : P(I - P)x = \mathbf{0} \implies \text{ran}(I - P) \subseteq \text{null}(P) \quad (1.1.2)$$

$$\forall x \in \text{null}(P) : Px = \mathbf{0} \implies (I - P)x = x \implies x \in \text{ran}(I - P) \quad (1.1.3)$$

$$\implies \text{ran}(I - P) = \text{null}(P) \quad (1.1.4)$$

□

This consequence states the fact that any vector x can be represented as $x = Px + (I - P)x$, and every projector will be defined via a subspace equals to the range of $I - P$ and P .

1.1.1 Orthogonal Projector

An orthogonal projector is a projector such that:

Definition 2.

$$\text{null}(P) \perp \text{ran}(P) \quad (1.1.5)$$

This property is in fact, very special. A good example of an orthogonal projector would be the Householder Reflector Matrix. Or just any $\hat{u}\hat{u}^H$ where \hat{u} is being an unitary vector. For convenience of proving, assume subspace $M = \text{ran}(P)$. Consider the following lemma:

Lemma 1.

$$\text{null}(P^H) = \text{ran}(P)^\perp \quad (1.1.6)$$

$$\text{null}(P) = \text{ran}(P^H)^\perp \quad (1.1.7)$$

Using (1.1.4) and consider the proof:

Proof.

$$\langle P^H x, y \rangle = \langle x, Py \rangle \quad (1.1.8)$$

$$\forall x \in \text{null}(P^H), y \in \mathbb{C}^n \quad (1.1.9)$$

$$\implies \langle P^H x, y \rangle = 0 = \langle x, Py \rangle \quad (1.1.10)$$

$$\implies \text{null}(P^H) \perp \text{ran}(P) \quad (1.1.11)$$

$$\forall y \in \text{null}(P), x \in \mathbb{C}^n : \quad (1.1.12)$$

$$\langle x, Py \rangle = 0 = \langle P^H x, y \rangle \quad (1.1.13)$$

$$\implies \text{ran}(P^H) \perp \text{null}(P) \quad (1.1.14)$$

□

Proposition 1. A projector is orthogonal iff it's Hermitian.

Proof. \Leftarrow Assuming the matrix is Hermitian and it's a projector, then we wish to prove that it's an orthogonal projector. Let's recall:

$$\text{null}(P^H) = \text{ran}(P)^\perp \quad (1.1.15)$$

$$\text{null}(P) = \text{ran}(P^H)^\perp \quad (1.1.16)$$

Substituting $P^H = P$, we have $\text{null}(P) = \text{ran}(P)^\perp$, Which is the definition of Orthogonal Projector. Therefore, P is an orthogonal projector by the definition of the projector.

For the \implies direction, we assume that P is an Orthogonal Projector, then we wish to show that it's also Hermitian. Observe that P^H is also a projector because $(P^H)^2 = (P^2)^H$. Then, using the definition of orthogonal projector:

$$\text{null}(P) \perp \text{ran}(P) \quad (1.1.17)$$

$$\text{null}(P^H) \perp \text{ran}(P^H) \quad (1.1.18)$$

Notice that using above statement together with Lemma 1 means $\text{null}(P) = \text{ran}(P)^\perp = \text{null}(P^H)$, and then $\text{ran}(P) = \text{null}(P)^\perp = \text{ran}(P^H)$. Therefore, P^H is an projector such that: $\text{ran}(P) = \text{ran}(P^H) \wedge \text{null}(P) = \text{null}(P^H)$. The range and null space of P^H and P is the same therefore P has to be Hermitian. □

1.1.2 Oblique Projector

An oblique projector a projector but not an orthogonal projector, and vice versa. It's a projector that satisfies the following conditions:

Definition 3.

$$Px \in M \quad (I - P)x \perp L \quad \text{where: } M \neq L \quad (1.1.19)$$

An orthogonal projector is the case when the subspace $M = L$.

A famous example of an orthogonal projector is QQ^H where Q is an Unitary Matrix. This is a Hermitian Matrix and it's idempotent, making it an orthogonal projector.

1.1.3 Projector Geometric Intuitions

A projector describes a given vector using some elements from another basis. The oblique projector creates a light sources in the form of the subspace L and it shoots parallel light rays in the direction orthogonal to L , crossing vectors and projecting their shadow onto subspace M .

1.1.4 Projector as a 2-Norm Minimizer

An orthogonal projector always reduce the 2 norm of a vector. Given any subspace M , we can create a basis of vectors packing into the matrix A , then P_M as a projector onto the basis M as an example can be: $A(AA^T)^{-1}A^T$. Let's consider the claim:

$$\|P_M x\|^2 \leq \|x\|^2 \quad (1.1.20)$$

Proof. For notational convenience, we simply denotes P_M using P .

$$x = Px + (I - P)x \quad (1.1.21)$$

$$\|x\|^2 = \|Px\|^2 + \|(I - P)x\|^2 \quad (1.1.22)$$

$$\|x\|^2 \geq \|Px\|^2 \quad (1.1.23)$$

□

Using this property of the Orthogonal Projector, we consider the following minimizations problem:

$$\min_{x \in M} \|y - x\|_2^2 = \|y - Py\|_2^2 \quad (1.1.24)$$

Proof:

$$\|y - x\|_2^2 = \|y - Py + Py - x\|_2^2 \quad (1.1.25)$$

$$\|y - x\|_2^2 = \|y - Py\|_2^2 + \|Py - x\|_2^2 \quad (1.1.26)$$

$$\implies \|y - Py\|_2^2 \leq \|y - x\|_2^2 \quad (1.1.27)$$

That concludes the proof. Observe that, $y - Py \perp M$ and $Py - x \in M$ because $Py, x \in M$, which allows us to split the norm of $y - x$ into 2 components. In addition using the fact that the projector is orthogonal. That concludes the proof.

1.2 Subspace Projection Methods

Let \mathcal{K}, \mathcal{L} be subspaces where candidates solutions are chosen and residuals are orthogonalized against. Under the idea case the 2 subspaces spans all dimensions, and it's able to approximate all solutions and forcing the residual vector $(b - Ax)$ to be zero. This is a description of this framework:

$$\tilde{x} \in x_0 + \mathcal{K} \text{ s.t: } b - A\tilde{x} \perp \mathcal{L} \quad (1.2.1)$$

it looks for an x in the affine linear subspace \mathcal{K} such that it's perpendicular to the subspace \mathcal{L} , or, equivalently, minimizing the projection onto the subspace \mathcal{L} . One interpretation of it is an projection of residual onto the basis that is orthogonal to \mathcal{L} .

Sometimes, for convenience and the exposition and exposing hidden connections between ideas, the above conditions can be expressed using matrix.

$$\text{Let } V \in \mathbb{C}^{n \times m} \text{ be a basis for: } \mathcal{K} \quad (1.2.2)$$

$$\text{Let } W \in \mathbb{C}^{n \times m} \text{ be a basis for: } \mathcal{L} \quad (1.2.3)$$

We can then make use of (1.3.1) and express it in the form of:

$$\tilde{x} = x^{(0)} + Vy \quad (1.2.4)$$

$$b - A\tilde{x} \perp \text{ran}(W) \quad (1.2.5)$$

$$W^T(b - A\tilde{x} - AVy) = \mathbf{0} \quad (1.2.6)$$

$$W^T r^{(0)} - W^T AVy = \mathbf{0} \quad (1.2.7)$$

$$W^T AVy = W^T r^{(0)} \quad (1.2.8)$$

1.2.1 Prototype Algorithm

And from here, we can define a simple prototype algorithm using this framework.

While not converging :

Increase Span for: \mathcal{K}, \mathcal{L}

Choose: V, W for \mathcal{K}, \mathcal{L}

$$r := b - Ax \quad (6)$$

$$y := (W^T AV)^{-1} W^T r$$

$$x := x + Vy$$

Each time, we increase the span of the subspace \mathcal{K}, \mathcal{L} , which gives us more space to choose the solution x , and more space to reduce the residual vector r . This idea is incredibly flexible, and we will see in later part where it reduces to a more concrete algorithm. Finally, when $\mathcal{K} = \mathcal{L}$, this is referred to as a Petrov Galerkin's Conditions.

1.2.2 Energy Norm Minimization using Gradient

Other times, iterative method will choose to build up a subspace for each step with a subspace generator, and build up the solution on this expanding subspace, but with the additional objective of minimizing the residual under some norm. Assuming that the vector $x \in x_0 + \mathcal{K}$, and we want to minimize the residual under a norm induced by positive definite operator B . Let it be the case that the columns of matrix K span subspace \mathcal{K} with $\dim(\mathcal{K}) = k$, then one may consider using gradient as a more direct approach instead of projector.

$$\min_{x \in x_0 + \mathcal{K}} \|b - Ax\|_B^2 \quad (1.2.9)$$

$$= \min_{w \in \mathbb{R}^k} \|b - A(x_0 + Kw)\|_B^2 \quad (1.2.10)$$

$$= \min_{w \in \mathbb{R}^k} \|r_0 - AKw\|_B^2 \quad (1.2.11)$$

We take the derivative of it and set the derivative to zero. We skip the proof that the derivative of $\nabla_x [\frac{1}{2}\|x\|_A^2] = Ax$, and for a crash course on derivative, the $\nabla_x[f(Ax)] = A^T\nabla[f(x)]$.

$$\nabla_w [\|r_0 - AKx\|_B^2] = \mathbf{0} \quad (1.2.12)$$

$$(AK)^T B(r_0 - AKx) = \mathbf{0} \quad (1.2.13)$$

$$(AK)^T Br_0 - (AK)^T BAKx = \mathbf{0} \quad (1.2.14)$$

$$(AK)^T Br_0 = (AK)^T BAKx \quad (1.2.15)$$

The above formulation is tremendously powerful. I used gradient instead of projector for the simplicity of the argument. One can derive the same using orthogonal projector to minimize the 2 norm, but the math is bit more tedious. However, this minimization objective is minimizing the residual, which is fine for deriving subspace methods such as the GMRes, or the Minres and Orthomin, however, for the sake of the conjugate gradient, we have to consider the alternative. Let's this be a proposition that we proceed to prove.

Proposition 2 (Conditions for Minimum Error Under Energy Norm). Here, we let matrix B be positive definite so it can induce a norm, we let K be a matrix whose columns forms a basis for \mathcal{K} , we let e_k denotes the error, given by: $A^{-1}b - x_k$, and we let r_k denotes the residual given as $b - Ax_k$.

$$\min_{x_k \in x_0 + \mathcal{K}} \|A^{-1}b - x\|_B^2 \iff K^T B e_k - K^T B K w = \mathbf{0} \quad (1.2.16)$$

Next, we proceed to prove it and explain its interpretations and importance:

Proof.

$$\min_{x \in x_0 + \mathcal{K}} \|A^{-1}b - x\|_B^2 = \min_{x \in \mathbb{R}} \|A^{-1}b - x_0 - Kw\|_B^2 \quad (1.2.17)$$

$$= \min_{x \in \mathbb{R}^k} \|e_k - Kw\|_B^2 \quad (1.2.18)$$

To attain the minimum of the norm, we take the derivative and set it to be zero, giving us:

$$\mathbf{0} = \nabla_w [\|e_k - Kw\|_B^2] \quad (1.2.19)$$

$$= \nabla_w [e_k - Kw]^T B (e_k - Kw) \quad (1.2.20)$$

$$= 2K^T B (e_k - Kw) \quad (1.2.21)$$

$$\implies K^T B e_k - K^T B K w = \mathbf{0} \quad (1.2.22)$$

□

This conditions is implicitly describing the objective of a Preconditioned Conjugate Gradient algorithm, where B is the M^{-1} matrix, however this discussion right now it's a digression. Instead let's set B to A , so that it's equivalent to the Energy Norm minimization of Conjugate Gradient, giving us this conditions:

$$K^T A A^{-1} r_k - K^T A K w = \mathbf{0} \quad (1.2.23)$$

$$K^T r_k - K^T A K w = \mathbf{0} \quad (1.2.24)$$

Here, we just made the substitution of $e_k = A^{-1}r_k$, and $B = A$. Later, we will see how this condition is linked to the idea of an Oblique Projector, similar to how an Orthogonal Projector is able to minimize the 2-Norm of the residual.

1.3 Krylov Subspace

A Krylov Subspace is a sequence basis paramaterized by A , an linear operator, v an initial vector, and k , which basis in the sequece of basis that we are looking at.

Definition 4 (Krylov subspace).

$$\mathcal{K}_k(A|b) = \text{span}(b, Ab, A^2b, \dots A^{k-1}b) \quad (1.3.1)$$

Please immediately observe that from the definition we have:

$$\forall v : \mathcal{K}_1(A|v) \subseteq \mathcal{K}_2(A|v) \subseteq \mathcal{K}_3(A|v) \dots \quad (1.3.2)$$

Please also observe that, every element inside of krylov subspace generated by matrix A , and an initial veoctr v can be represented as a polynomial of matrix A multiplied by the vector v and vice versa.

$$\forall x \in \mathcal{K}_k(A|v) \exists w : p_k(A|w)v = x \quad (1.3.3)$$

We use $p_k(A|w)$ to denotes a matrix polynomial with coefficients $w \in \mathcal{K}_j$, where w is a vector. No proof this is trivial. Take note that, we can change the field of where the scalar w_i is coming from, but for discussion below, \mathbb{R}, \mathbb{C} doesn't matter and won't change the results so we stick to \mathbb{R} and we let v, A be real vectors and matrices so it's consistent.

$$p_k(A|w)v = \sum_{j=0}^{k-1} w_j A^j v \quad (1.3.4)$$

1.3.1 The Grade of a Krylov Subspace

The most important porperty of the subspace is the idea of grade denoted as: $\text{grade}(A|v)$, indicating when the Krylov Subspace of A wrt to v stops expanding after a certain size. To show this idea, we consider 3 propositions about Krylov Subspace. And then we show that ir results in the existence of a grade of the Krylov Subspace, and then proceed to prove these 3 propositions we made.

Proposition 3.

$$\exists 1 \leq k \leq m+1 : \mathcal{K}_k(A|v) = \mathcal{K}_{k+1}(A|v) \quad (1.3.5)$$

There exists an natural number between 1 and $m+1$ such that, the successive krylov subspace span the same space asthe previous one.

Proposition 4.

$$\exists \min k \text{ s.t: } \mathcal{K}_k(A|v) = \mathcal{K}_{k+1}(A|v) \implies \mathcal{K}_k(A|v) \text{ is Lin Ind} \wedge \mathcal{K}_{k+1}(A|v) \text{ is Lin Dep.} \quad (1.3.6)$$

There eixsts a minimum such k where the immediate next krylov subspace is linear dependent.

Proposition 5.

$$\mathcal{K}_k(A|v) \text{ Lin Dep} \implies \mathcal{K}_{k+1}(A|v) = \mathcal{K}_k(A|v) \quad (1.3.7)$$

if the k krylov subspace is linear dependent, then it stops expanding and the successive krylov subspace spans the same space.

Theorem 1 (Existence of Grade for a Krylov Subspace). Let k be the minumum number when the krylov subspace stops expanding, then all successive krylov subspace spand the same space. $\mathcal{K}_k(A|v) = \mathcal{K}_{k+j}(A|v) \forall j \geq 0$. The number k is regarded as the grade of krylov subspace wrt to v denoted using $\text{grade}(A|v)$.

Proposition 2, 3 ensures that there exists a term in the sequence of krylov subspace stops expanding, and when that happens all subsequent Krylov Subspace will span the same subspace, this is by Proposition 4.

Next, let's consider the proof of theorem 1 by proving all proposition 2, 3, and 4.

Proposition 2. For notational simplicity, \mathcal{K}_k now denotes $\mathcal{K}_k(A|v)$. Let's start the considerations from the definition of the Krylov Subspace:

$$\forall k : \mathcal{K}_k \subseteq \mathcal{K}_{k+1} \implies \dim(\mathcal{K}_k) \leq \dim(\mathcal{K}_{k+1}) \quad (1.3.8)$$

$$\mathcal{K}_{k+1} \setminus \mathcal{K}_k = \text{span}(A^k v) \quad (1.3.9)$$

$$\implies \dim(\mathcal{K}_{k+1}) - \dim(\mathcal{K}_k) \leq 1 \quad (1.3.10)$$

Therefore, the dimension of the successive krylov subspace forms a sequence of positive integer that is monotonically increasing. By the Cayley's Hamilton's theorem (will be stated later), the sequence is bounded by m , since there are $m + 1$ terms, it must be the case that at least 2 of the krylov subspace has the same dimension (And the earliest such occurance will exist), implying the the fact that the new added vector from k to $k + 1$ is in the span of the previous subspace. \square

Proposition 4.

$$\mathcal{K}_k \text{ Lin Dep} \quad (1.3.11)$$

$$\implies \exists w_k \neq \mathbf{0} : p_k(A|w_k^+)v = \mathbf{0} \quad (1.3.12)$$

$$\implies Ap_k(A|w_k^+)v = \mathbf{0} \quad (1.3.13)$$

$$p_{k+1}(A|[0 \ (w_k^+)^T]) = \mathbf{0} \quad (1.3.14)$$

$$\mathcal{K}_{k+1} \text{ is Lin Dep} \quad (1.3.15)$$

The recurrence of multiplying by A allows the krylov subspace to grow and the new bigger subspace will contain the previous one. Therefore inheriting the linear dependence, we use the idea of matrix polynomial for the proof. \square

Proposition 3. Assuming that prop 2, 4 are true. prop 3 implies the existence of the smallest such k . For contradiction, we only have one case to assume, that is \mathcal{K}_k and \mathcal{K}_{k+1} are linear dependence. Then \mathcal{K}_k is either Linear Dependence, or Independence.

If \mathcal{K}_{k-1} is linear dependence, then by (3) $\mathcal{K}_{k-1} = \mathcal{K}_k$, hence k is not the minimum. Else assume \mathcal{K}_{k-1} is linear independence, however \mathcal{K}_k is linear dependence, and $\mathcal{K}_k \setminus \mathcal{K}_{k-1} = \text{span}(A^{k-1}v)$; therefore, $A^{k-1}v$ is in the span of \mathcal{K}_{k-1} , hence $\mathcal{K}_{k-1} = \mathcal{K}_k$, contradicting again that k is the minimum such k . \square

1.4 Useful Theorems and Mathematical Entities

1.4.1 Minimal Polynomial of a Matrix

Definition 5. A minimal polynomial is a polynomial $p_k(x)$ with degree k that is monic such that $p_k(A) = \mathbf{0}$.

1.4.2 Cauchy Interlace Theorem

The Cauchy's interlace theorem describes the relations of eigenvalues between the submatrix of a Symmetric tridiagonal matrix and the matrix itself. In our case, let $T_{k-1} \in \mathbb{R}^{(k-1) \times (k-1)}$ be the principal sub-matrix to the matrix T_k , then between every eigenvalue of T_k , there must exist an eigenvalue of T_{k-1} between them.

1.4.3 Cayley Hamilton's Theorem

Theorem 2. A matrix satisfies its own characteristic equation, let $p(x)$ be the characteristic polynomial for the matrix A , then $p(A) = \mathbf{0}$.

The Cayley's Hamilton's Theorem is important in the sense that, a direct consequence is the termination conditions for all Krylov Subspace Methods (regardless of initial guess vectors). It's saying that all Krylov Subspace methods will terminate at step $n + 1$ at most, if n is the size of the operator. However, the more important fact is that, when it terminates, the solution x is a weighted sum of the Krylov Subspace vectors and the weights are related to the characteristic polynomial of the matrix.

1.4.4 The Chebyshev Polynomial

The Chebyshev polynomial is an orthogonal polynomial under a weighted space; it has amazing properties, for our sake, we focus on the Type T Chebyshev Polynomial which has the property of minimizing the infinity norm over a closed interval:

$$T_k(x) = \arg \min_{p(x)} \{ \|p(x)\|_\infty : \forall x \in [-1, 1] \} \quad (1.4.1)$$

This theorem is later used as a pivotal tool for the analysis of floating point error for the conjugate gradient algorithm.

1.5 Deriving Conjugate Gradient from First Principles

1.5.1 CG Objective and Framework

We introduce the algorithm as an attempt to minimize the energy norm of the error for a linear equation $Ax = b$, here we make the assumptions:

- 1) The matrix A is symmetric semi-positive definite.
- 2) Further assume another matrix $P_k = [p_0 \ p_1 \ \cdots \ p_{k-1}]$ as a matrix whose columns are a basis.

$$\min_{w \in \mathbb{R}^k} \|A^{-1}b - (x_0 + P_k w)\|_A^2 \iff P_k^T r_0 = P_k^T A P_k w \quad (1.5.1)$$

Refer back to (1.4) for how to deal with the above minimization objective. Using the matrix form for the Petrov Galerkin Conditions where W, V are both P_k , we reformulate the Norm Minimizations conditions under the framework of Petrov Galarkin conditions:

$$\text{choose: } x \in x_0 + \text{ran}(P_k) \text{ s.t: } b - Ax \perp \text{ran}(P_k) \quad (1.5.2)$$

Take note that the link between a norm minimization and an equivalent subspace Orthogonality conditions don't guarantee to happen for other subspace projector methods, for example the FOM and Bi-Lanczos Methods are orthogonalizations method that doesn't directly link to a norm minimization objective.

To solve for w , we wish to make $P_k^T A P_k$ to be an easy-to-solve matrix. Let the easy-to-solve matrix to be a diagonal matrix and hence we let P_k to be a *matrix whose columns are A-Orthogonal vectors*.

$$P_k^T A P_k = D_k \text{ where: } (D_k)_{i,i} = \langle p_{i-1}, A p_{i-1} \rangle \quad (1.5.3)$$

$$P_k r_0 = P_k^T A P_k w = D_k w \quad (1.5.4)$$

$$w = D_k^{-1} P_k^T r_0 \quad (1.5.5)$$

The idea here is: Accumulating vectors p_j into the matrix P_k and then iterative improve the solution x_k , by reducing the error denote as e_k which is defined as $A^{-1}b - x_k$. Then, we can derive the following expression for the solution at step k x_k and the residual at step $r_k = b - A x_k$ for the algorithm:

$$\begin{cases} x_k = x_0 + P_k D_k^{-1} P_k^T r_0 \\ r_k = r_0 - A P_k D_k^{-1} P_k^T r_0 \\ P_k^T A P_k = D_k \end{cases} \quad (1.5.6)$$

Let this algorithm be the prototype.

1.5.2 Using the Projector

Here, we consider the above prototype algorithm. Please observe that $A P_k D_k^{-1} P_k$ is a projector, and so is $P_k D_k^{-1} P_k^T A$.

Proof.

$$A P_k D_k^{-1} P_k^T (A P_k D_k^{-1} P_k^T) = A P_k D_k^{-1} P_k^T A P_k D_k^{-1} P_k^T \quad (1.5.7)$$

$$= A P_k D_k^{-1} D_k D_k^{-1} P_k^T \quad (1.5.8)$$

$$= A P_k D_k^{-1} P_k^T \quad (1.5.9)$$

$$P_k D_k^{-1} P_k^T A (P_k D_k^{-1} P_k^T A) = P_k D_k^{-1} D_k D_k^{-1} P_k^T A \quad (1.5.10)$$

$$= P_k D_k^{-1} P_k^T A \quad (1.5.11)$$

□

Both matrices are indeed projectors. Please take note that they are not Hermitian, which would mean that they are not orthogonal projector, hence, oblique projectors. For notational convenience, we denote $\bar{P}_k = P_k D_k^{-1} P_k^T$; then these 2 projectors are:

$$AP_k D_k^{-1} P_k^T = A\bar{P}_k \quad (1.5.12)$$

$$P_k D_k^{-1} P_k^T A = \bar{P}_k A \quad (1.5.13)$$

One immediate consequence is:

$$\text{ran}(I - A\bar{P}_k) \perp \text{ran}(P_k) \quad (1.5.14)$$

$$\text{ran}(I - \bar{P}_k A) \perp \text{ran}(AP_k) \quad (1.5.15)$$

Proof.

$$P_k^T (I - A\bar{P}_k) = P_k^T - P_k^T A\bar{P}_k \quad (1.5.16)$$

$$= P_k^T - D_k D_k^{-1} P_k^T \quad (1.5.17)$$

$$= \mathbf{0} \quad (1.5.18)$$

$$(AP_k)^T (I - \bar{P}_k A) = P_k^T A - P_k^T A\bar{P}_k A \quad (1.5.19)$$

$$= P_k^T A - P_k^T AP_k D_k^{-1} P_k^T A \quad (1.5.20)$$

$$= P_k^T A - P_k^T A \quad (1.5.21)$$

$$= \mathbf{0} \quad (1.5.22)$$

□

Using the properties of the oblique projector, we can proof 2 facts about this simple norm minimization method we developed:

Proposition 6 (Residuals are Orthogonal to P_k).

$$r_k = r_0 - A\bar{P}_k r_0 = (I - A\bar{P}_k) r_0 \quad (1.5.23)$$

$$\implies r_k \perp \text{ran}(P_k) \quad (1.5.24)$$

Proposition 7 (Generating A Orthogonal Vectors). Given any set of basis vector, for example $\{u_k\}_{i=0}^{n-1}$, one can generate a set of A -Orthogonal vectors from it. More specifically:

$$p_k = (I - \bar{P}_k A) u_k \quad (1.5.25)$$

$$\text{span}(p_k) \perp \text{ran}(AP_k) \quad (1.5.26)$$

For above propositions, we used the immediate consequence of the range of these oblique projectors.

1.5.3 Assisted Conjugate Gradient

So far, we have this particular scheme of solving the optimization problem, coupled with the way to computing the solution x_k at each step, and the residual at each step, while also getting the residual vector at each step too. However, it would be great if we can accumulate

on the same subspace P_k and look for a chance to reuse the computational results from the previous iterations of the algorithm:

$$\begin{cases} x_k = x_0 + \bar{P}_k r_0 \\ r_k = (I - A\bar{P}_k)r_0 \\ P_k^T A P_k = D_k \\ \bar{P}_k = P_k D_k^{-1} P_k^T \\ p_k = (I - \bar{P}_k A)u_k \quad \{u_i\}_{i=0}^{n-1} \text{ is a Basis} \end{cases} \quad (1.5.27)$$

With the assistance of a set of basis vector that span the whole space, this algorithm is possible to achieve the objective. Take note that we can accumulate the solution for x_k accumulatively, instead of computing the whole projector process, we have the choice to update it recursively as the newest p_k vector is introduced at that step. Let's Call this formulation of the algorithm: *Assisted Conjugate Gradient*.

1.5.4 Properties of Assisted Conjugate Gradient

Here we setup several useful lemma and propositions that can derive the short recurrences of A-Orthogonal vectors

Proposition 8.

$$p_{k+j}^T r_k = p_{k+j}^T r_0 \quad \forall 0 \leq j \leq n - k \quad (1.5.28)$$

$$p_{k+j}^T r_k = p_k^T (I - A\bar{P}_k)r_0 \quad (1.5.29)$$

$$= (p_{k+j}^T - p_{k+j}^T A\bar{P}_k)r_0 \quad (1.5.30)$$

$$= p_{k+j}^T r_0 \quad (1.5.31)$$

This made the recurrence between successive residual from the ACG possible.

Next, we wish to use this property to find out a recurrences for the residuals of ACG, and here is how we do it:

$$r_k - r_{k-1} = r_0 - A\bar{P}_k r_0 - (r_0 - A\bar{P}_{k-1} r_0) \quad (1.5.32)$$

$$= A\bar{P}_k r_0 - A\bar{P}_{k-1} r_0 \quad (1.5.33)$$

$$= -A p_{k-1} \frac{\langle p_{k-1}, r_0 \rangle}{\langle p_{k-1}, A p_{k-1} \rangle} \quad (1.5.34)$$

$$\implies x_k - x_{k-1} = p_{k-1} \frac{\langle p_{k-1}, r_0 \rangle}{\langle p_{k-1}, A p_{k-1} \rangle} \quad (1.5.35)$$

$$\text{def: } a_{k-1} := \frac{\langle p_{k-1}, r_0 \rangle}{\langle p_{k-1}, A p_{k-1} \rangle} = \frac{\langle p_{k-1}, r_{k-1} \rangle}{\langle p_{k-1}, A p_{k-1} \rangle} \quad (1.5.36)$$

We define the value of a_{k-1} , and in above, we have 2 equivalent representation. Please take note that, Proposition still remains true for the ACG algorithm we just developed here.

1.5.5 Residual Assisted Conjugate Gradient

Now, consider the case where, the set of basis vector: $\{u\}_{i=0}^{n-1}$ to be the residual vector generated from the ACG itself. Then there are a series of new added lemmas that are true. However, this is where things started to get exciting, because a short recurrence for p_k during each iteration arised and residuals are all orthgonal. We wish to proceed to prove that part.

Lemma 2.

$$\langle p_{k+j}, Ap_k \rangle = \langle r_k, Ap_{k+j} \rangle = \langle p_{k+j}, Ar_k \rangle \quad \forall 0 \leq j \leq n - k \quad (1.5.37)$$

Proof.

$$p_{k+j}Ap_k = p_{k+j}^T Ar_k - p_{k+j}^T A\bar{P}_k Ar_k \quad \forall 0 \leq j \leq n - k \quad (1.5.38)$$

$$= p_{k+j}^T Ar_k \quad (1.5.39)$$

$$\langle p_{k+j}, Ap_k \rangle = \langle r_k, Ap_{k+j} \rangle = \langle p_{k+j}, Ar_k \rangle \quad (1.5.40)$$

□

Lemma 3.

$$\langle r_k, p_k \rangle = \langle r_k, r_k \rangle \quad (1.5.41)$$

Proof.

$$\langle r_k, p_k \rangle = \langle r_k, p_k \rangle \quad (1.5.42)$$

$$= \langle r_k, r_k \rangle - \langle r_k, \bar{P}_k Ar_k \rangle \quad (1.5.43)$$

$$= \langle r_k, r_k \rangle \quad (1.5.44)$$

From the first line to the second line, we make use of the definition proposed. □

Proposition 9 (Residual Assisted CG Generates Orthogonal Residuals).

$$\langle r_k, r_j \rangle = 0 \quad \forall 0 \leq j \leq k - 1 \quad (1.5.45)$$

Let this above claim be inductively true then consider the following proof:

Proof.

$$r_{k+1} = r_k - a_k Ap_k \quad (1.5.46)$$

$$\implies \langle r_{k+1}, r_k \rangle = \langle r_k, r_k \rangle - a_k \langle r_k, Ap_k \rangle \quad (1.5.47)$$

$$= \langle r_k, r_k \rangle - \frac{\langle r_k, r_k \rangle}{\langle p_k, Ap_k \rangle} \langle r_k, Ap_k \rangle \quad (1.5.48)$$

$$= 0 \quad (1.5.49)$$

The first line is from the recurrence of ACG residuals, and then next we make use of the updated definition for a_k . Next we consider:

$$p_j = (I - \bar{P}_j A) r_j \quad \forall 0 \leq j \leq k - 1 \quad (1.5.50)$$

$$r_j = p_j + \bar{P}_j Ar_j \quad (1.5.51)$$

$$r_k = (I - A\bar{P}_k) P_0 \quad (1.5.52)$$

$$r_k \perp \text{ran}(P_k) \implies \langle r_k, r_j \rangle = \langle r_k, p_j + \bar{P}_j Ar_j \rangle = 0 \quad (1.5.53)$$

Here we again make use of the projector $I - A\bar{P}_k$. The base case of the argument is simple, because $p_0 = r_0$, and by the property of the projector, $\langle r_1, r_0 \rangle = 0$. The theorem is now proven. \square

Proposition 10 (RACG Recurrences).

$$p_k = r_k + b_{k-1}p_{k-1} \quad b_{k-1} = \frac{\|r_k\|_2^2}{\|r_{k-1}\|_2^2} \quad (1.5.54)$$

The proof is direct and we start with the definition of ACG, which is given as:

Proof.

$$p_k = (I - \bar{P}_k A)r_k \quad (1.5.55)$$

$$r_k - \bar{P}_k A r_k = r_k - P_k D_k^{-1} P_k^T A r_k \quad (1.5.56)$$

$$= r_k - P_k D_k^{-1} (A P_k)^T r_k \quad (1.5.57)$$

Observe that the term $(A P_k)^T$ can be expanded and we can make use of the Symmetric Property of the operator A_k .

$$(A P_k)^T r_k = \begin{bmatrix} \langle p_0, A r_k \rangle \\ \langle p_1, A r_k \rangle \\ \vdots \\ \langle p_{k-1}, A r_k \rangle \end{bmatrix} \quad (1.5.58)$$

Next, we can make use of Lemma 2 to get rid of $A r_k$. Please consider:

$$(A P_k)^T r_k = \begin{bmatrix} \langle p_0, A r_k \rangle \\ \langle p_1, A r_k \rangle \\ \vdots \\ \langle p_{k-1}, A r_k \rangle \end{bmatrix} \quad (1.5.59)$$

The second line is using the property that the matrix A is symmetric, the third line is using the recurrence of the residual of ACG, and the last line is true for all $0 \leq j \leq k-2$ by the orthogonality of the residual proved in Claim 1. Therefore we have:

$$(A P_k)^T r_k = \begin{bmatrix} \langle p_0, A r_k \rangle \\ \langle p_1, A r_k \rangle \\ \vdots \\ \langle p_{k-1}, A r_k \rangle \end{bmatrix} = a_{k-1}^{-1} \langle r_k, (r_{k-1} - r_k) \rangle \xi_k \quad (1.5.60)$$

Take note that the vector ξ_k is the k th standard basis vector in \mathbb{R}^k . And using this we can

simplify the expression for p_k into:

$$p_k = r_k - P_k D_k^{-1} (AP_k)^T r_k \quad (1.5.61)$$

$$= r_k - P_k D_k^{-1} a_{k-1}^{-1} (\langle r_k, (r_{k-1} - r_k) \rangle) \xi_k \quad (1.5.62)$$

$$= r_k - \frac{a_{k-1}^{-1} \langle -r_k, r_k \rangle}{\langle p_{k-1}, Ap_{k-1} \rangle} p_k \quad (1.5.63)$$

$$= r_k + \frac{a_{k-1}^{-1} \langle r_k, r_k \rangle}{\langle p_{k-1}, Ap_{k-1} \rangle} p_k \quad (1.5.64)$$

$$= r_k + \left(\frac{\langle r_{k-1}, r_{k-1} \rangle}{\langle p_{k-1}, Ap_{k-1} \rangle} \right)^{-1} \frac{\langle r_k, r_k \rangle}{\langle p_{k-1}, Ap_{k-1} \rangle} p_k \quad (1.5.65)$$

$$= r_k + \frac{\langle r_k, r_k \rangle}{\langle r_{k-1}, r_{k-1} \rangle} p_k \quad (1.5.66)$$

We make use of the definition for a_{k-1} in figured out for the *ACG* algorithm. At this point, we have proven the RACG recurrences. \square

Up until this point we have proven the usual version of conjugate gradient, we started with the minimizations objective and the properties of P_k , then we define a recurrences for the residual (Simultaneously the solution x_k), and the A-Orthogonal vectors using a basis as assistance for the generations process. Next, we make the key changes of the assistance basis, making it equal to the set of residuals vector generated from the algorithm itself; after some proof, we uncovered the exact same parameters found in most of the definitions of the CG algorithm, which we refers to as Residual Assisted Conjugate Gradient. Here we proposed the RACG:

Definition 6 (RACG).

$$p^{(0)} = b - Ax^{(0)} \quad (1.5.67)$$

$$\text{For } i = 0, 1, \dots \quad (1.5.68)$$

$$\begin{aligned} a_i &= \frac{\|r^{(i)}\|^2}{\|p^{(i)}\|_A^2} \\ x^{(i+1)} &= x^{(i)} + a_i p^{(i)} \\ r^{(i+1)} &= r^{(i)} - a_i A p^{(i)} \\ b_i &= \frac{\|r^{(j+1)}\|_2^2}{\|r^{(i)}\|_2^2} \\ p^{(i+1)} &= r^{(i+1)} + b_i p^{(i)} \end{aligned} \quad (1.5.69)$$

That is the algorithm, stated with all the iteration number listed as a super script inside of a parenthesis. Which is equivalent to what we have proven for the Residual Assisted Conjugate Gradient.

1.5.6 RACG and Krylov Subspace

The conjugate Gradient Algorithm is actually a residual assisted conjugate gradient, a special case of the algorithm we derived at the start of the excerpt. The full algorithm can be seen by

the short recurrence for the residual and the conjugation vector. This part is trivial. Next, we want to show the relations to the Krylov Subspace, which only occurs for the Residual Assisted Conjugate Gradient algorithm.

Proposition 11.

$$p_k \in \mathcal{K}_{k+1}(A|r_0) \quad (1.5.70)$$

$$r_k \in \mathcal{K}_{k+1}(A|r_0) \quad (1.5.71)$$

Proof. The base case is trivial and it's directly true from the definition of Residual Assisted Conjugate Gradient: $r_0 \in \mathcal{K}_1(A|r_0), p_0 = r_0 \in \mathcal{K}_1(A|r_0)$. Next, we inductively assume that $r_k \in \mathcal{K}_{k+1}(A|r_0), p_k \in \mathcal{K}_{k+1}(A|r_0)$, then we consider:

$$r_{k+1} = r_k - a_k A p_k \quad (1.5.72)$$

$$\in r_k + A\mathcal{K}_{k+1}(A|r_0) \quad (1.5.73)$$

$$\in r_k + \mathcal{K}_{k+2}(A|r_0) \quad (1.5.74)$$

$$r_k \in \mathcal{K}_{k+1}(A|r_0) \subseteq \mathcal{K}_{k+2}(A|r_0) \quad (1.5.75)$$

$$\implies r_{k+1} \in \mathcal{K}_{k+2}(A|r_0) \quad (1.5.76)$$

At the same time the update of p_k would asserts the property that:

$$p_{k+1} = r_{k+1} + b_k p_k \quad (1.5.77)$$

$$\in r_{k+1} + \mathcal{K}_{k+1}(A|r_0) \quad (1.5.78)$$

$$\in \mathcal{K}_{k+2}(A|r_0) \quad (1.5.79)$$

This is true because r_{k+1} is already a member of the expanded subspace $\mathcal{K}_{k+2}(A|r_0)$. And from this formulation of the algorithm, we can update the Petrov Galerkin's Conditions to be:

$$\text{choose: } x_k \in x_0 + \mathcal{K}_k(A|r_0) \text{ s.t. } r_k \perp \mathcal{K}_k(A|r_0) \quad (1.5.80)$$

Take note that, $\text{ran}(P_k) = \mathcal{K}_k(A|r_0)$ because the index starts with zero. The above formulations gives theoretical importance for the Conjugate Gradient Algorithm. \square

1.6 Arnoldi Iterations and Lanczos

In this section, we introduce another important algorithm: The Lanczos Algorithm. However, to give more context for the discussion, the Arnoldi iteration is considered as well and it's used to emphasize that Lanczos Iterations is just Arnoldi but with the matrix A being a symmetric matrix. Finally we make the link between Lanczos Iterations and Krylov Subspace, which will inevitably be linked back to RACG and plays an important role for the analysis of RACG.

1.6.1 The Arnoldi Iterations

We first define the Arnoldi Algorithm, and then we proceed to derive it using the idea of orthogonal projector. Next, we discuss a special case of the Arnoldi Iteration: the Lanczos

Algorithm, which is just Arnoldi applied to a symmetric matrix. And such algorithm will inherit the properties of the Arnoldi Iterations.

Before stating the algorithm, I would like to point out the interpretations of the algorithm and its relations to Krylov Subspace. Consider a matrix of Hessenberg Form:

$$\tilde{H}_k = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,k} \\ h_{1,2} & h_{2,2} & \cdots & h_{2,k} \\ & \ddots & & \vdots \\ & & h_{k,k-1} & h_{k,k} \\ & & & h_{k+1,k} \end{bmatrix} \quad (1.6.1)$$

We initialize the orthogonal projector with the vector q_1 , which is $q_1 q_1^H$, next, we apply the linear operator A on the current range of the projector: Aq_1 , then, we orthogonalize it against q . Let the projection of Aq_1 onto $I - q_1 q_1^H$ be $h_{1,2}q_2$, and let the projection onto $q_1 q_1^H$ be $h_{1,1}$. This completes the first column of H_k , we do this recursively. Please allow me to demonstrate:

$$(\tilde{H}_k)_{2,1}q_2 = (I - q_1 q_1^H)Aq_1 \quad (1.6.2)$$

$$(\tilde{H}_k)_{1,1}q_1 = q_1 q_1^H Aq_1 \quad (1.6.3)$$

$$Q_2 := [q_1 \quad q_2] \quad (1.6.4)$$

$$(\tilde{H}_k)_{3,2}q_3 = (I - Q_2 Q_2^H)Aq_2 \quad (1.6.5)$$

$$(\tilde{H}_k)_{1:2,2} = Q_2 Q_2^H Aq_2 \quad (1.6.6)$$

$$Q_3 := [q_1 \quad q_2 \quad q_3] \quad (1.6.7)$$

$$\vdots \quad (1.6.8)$$

$$Q_j := [q_1 \quad q_2 \quad \cdots \quad q_j] \quad (1.6.9)$$

$$(\tilde{H}_k)_{j+1,j}q_{j+1} = (I - Q_j Q_j^H)Aq_j \quad (1.6.10)$$

$$(\tilde{H}_k)_{1:j,j} = Q_j Q_j^H Aq_j \quad (1.6.11)$$

$$\vdots \quad (1.6.12)$$

$$Q_k := [q_1 \quad q_2 \quad \cdots \quad q_k] \quad (1.6.13)$$

$$(\tilde{H}_k)_{k+1,k}q_{k+1} = (I - Q_k Q_k^H)Aq_k \quad (1.6.14)$$

$$(\tilde{H}_k)_{1:k,k} = Q_k Q_k^H Aq_k \quad (1.6.15)$$

Reader please observe that Q_k is going to be orthogonal because how at the start, $q_1 q_1^H$ and $I - q_1 q_1^H$ is giving us an orthogonal subspace. As a consequence, we can express the recurrences of the subspace vector in matrix form:

$$AQ_k = Q_{k+1} \tilde{H}_k \quad (1.6.16)$$

$$Q_k^H A Q_k = H_k \quad (1.6.17)$$

And here, we explicitly define H_k to be the principal submatrix of \tilde{H}_k . Reader please immediately observe that, if A is symmetric, then it has to be the case that $Q_k^H A Q_k$ is also

symmetric, which will make H_k to be symmetric as well, which implies that H_k will be a Symmetric Tridiagonal Matrix. And under that assumption, we can develop the Lanczos Algorithm. Instead of orthogonalizing against all previous vectors, we have the option to simply orthogonalize against the previous q_k, q_{k-1} vector. And we can reuse the sub-diagonal elements for q_{k-1} ; giving us the Lanczos Algorithm.

1.6.2 Arnoldi Produces Orthogonal Basis for Krylov Subspace

One important observations reader should make about the idea of Arnoldi Iteration is that, during each iteration, the matrix Q_k spans the same range as $\mathcal{K}_k(A|q_1)$.

Proposition 12.

$$\text{ran}(Q_k) = \mathcal{K}_k(A|q_1) \quad (1.6.18)$$

Proof. The base case is simple: $q_1 \in \mathcal{K}_1(A|q_1)$, inductively assuming the proposition is true, using the polynomial property of Krylov Subspace we consider:

$$\begin{aligned} Q_k &\in \mathcal{K}_k(A|q_1) \\ \iff w_k^+ : \exists p_k(A|w_k^+)q_1 &= q_k \\ \implies Aq_k &= Ap_k(A|w_k^+)q_1 \in \mathcal{K}_{k+1}(A|w_k^+) \\ q_{k+1} &\in \mathcal{K}_{k+1}(A|q_1) \\ \implies \text{ran}(Q_{k+1}) &= \mathcal{K}_{k+1}(A|q_1) \end{aligned}$$

The Arnoldi Algorithm terminates if the value $h_{k+1,k}$ is set to be zero. This is the case because the normalization process is dividing by $h_{k+1,k}$ to get q_{k+1} . This only happens when $Aq_k \in \text{ran}(Q_k)$; because $h_{k+1,k}$ is given by the projector of $I - Q_k Q_k^H$ applied to Aq_k and the null space of this projector is $\text{ran}(Q_k)$, resulting in $h_{k+1,k} = 0$. \square

1.6.3 The Lanczos Iterations

Definition 7 (Lanczos Iterations).

$$\text{Given arbitrary: } q_1 \text{ s.t: } \|q_1\| = 1 \quad (1.6.19)$$

$$\text{set: } \beta_0 = 0 \quad (1.6.20)$$

$$\text{For } j = 1, 2, \dots \quad (1.6.21)$$

$$\begin{aligned} \tilde{q}_{j+1} &:= Aq_j - \beta_{j-1}q_{j-1} \\ \alpha_j &:= \langle q_j, \tilde{q}_{j+1} \rangle \\ \tilde{q}_{j+1} &\leftarrow \tilde{q}_{j+1} - \alpha_j q_j \\ \beta_j &= \|\tilde{q}_{j+1}\| \\ q_{j+1} &:= \tilde{q}_{j+1}/\beta_j \end{aligned} \quad (1.6.22)$$

Here, let it be the case that H_k is a Symmetric Tridiagonal Matrix with α_i on the diagonal, β_i on the sub and super diagonal; the lanczos is Arnoldi, but we make use of the symmetric properties to orthogonalize Aq_j against q_{j-1} using β_{j-1} , and in this case, each iteration only

consists of one vector inner product. Note that another equivalent algorithm wher I tweaked it to handle the base case of T_k being a 1×1 matrix can be phrased in the following way:

$$\begin{aligned}
&\text{Given arbitrary: } q_1 \text{ s.t: } \|q_1\| = 1 \\
&\alpha_1 := \langle q_1, Aq_1 \rangle \\
&\beta_0 := 0 \\
&\text{Memorize : } Aq_1 \\
&\text{For } j = 1 \cdots \\
&\quad \tilde{q}_{j+1} := Aq_j - \beta_{j-1}q_{j-1} \\
&\quad \tilde{q}_{j+1} \leftarrow \tilde{q}_{j+1} - \alpha_j q_j \\
&\quad \beta_j = \|\tilde{q}_{j+1}\| \\
&\quad q_{j+1} := \tilde{q}_{j+1}/\beta_j \\
&\quad \alpha_{j+1} := \langle q_{j+1}, Aq_{j+1} \rangle \\
&\quad \text{Memorize: } Aq_{j+1}
\end{aligned} \tag{1.6.23}$$

Often time, we refers the $k \times k$ symmetric tridiagonal matrix generated from Iterative Lanczos as T_k . Finally; I wish to make the following important remark about the algorithm for later use. Given a matrix A and an initial vector q_1 , The lanczos algorithm produces an irreducible Symmetric Tridiagonal Matrix that has unique eigenvalues. The proof for the fact that any Symmetric Tridiaogonal Matrices with Non-zeros on the sub/super diagonal must have unique non-zero eigenvalues is skipped. What we can immediate show here is the fact that Lanczos Algorithm will produce such a matrix.

Proposition 13. The Lanczos Iteration produces a Symmetric Tridiagonal Matrix that has no zero element on its super and sub-diagonal, and if β_k is zero, then the algorithm must terminates, and k would equal to $\text{grade}(A|q_1)$, the grade of the Krylov Subspace.

Proof. It's true because the β_k in the Lanczos is equivalent to $h_{k+1,k}$. It's been discussed previously that if $h_{k+1,1} = 0$ for the Arnoldi's Iteration, then the Krylov Subspace $\mathcal{K}_k(A|q_1)$ became an invariant subspace under A , and in that sense, the algorithm has to terminate due to a divides by zero error. \square

2	Analysis of Conjugate Gradient and Lanczos Iterations
2.1	Terminations Conditions of RACG
2.2	Terminations Conditions of The Lanczos Iterations
2.3	Convergence Rate of RACG under Exact Arithematic
2.4	From Conjugate Gradient to Lanczos
2.5	From Lanczos to Conjugate Gradient
3	Effects of Floating Point Arithematic
4	Appendix
5	Biliography