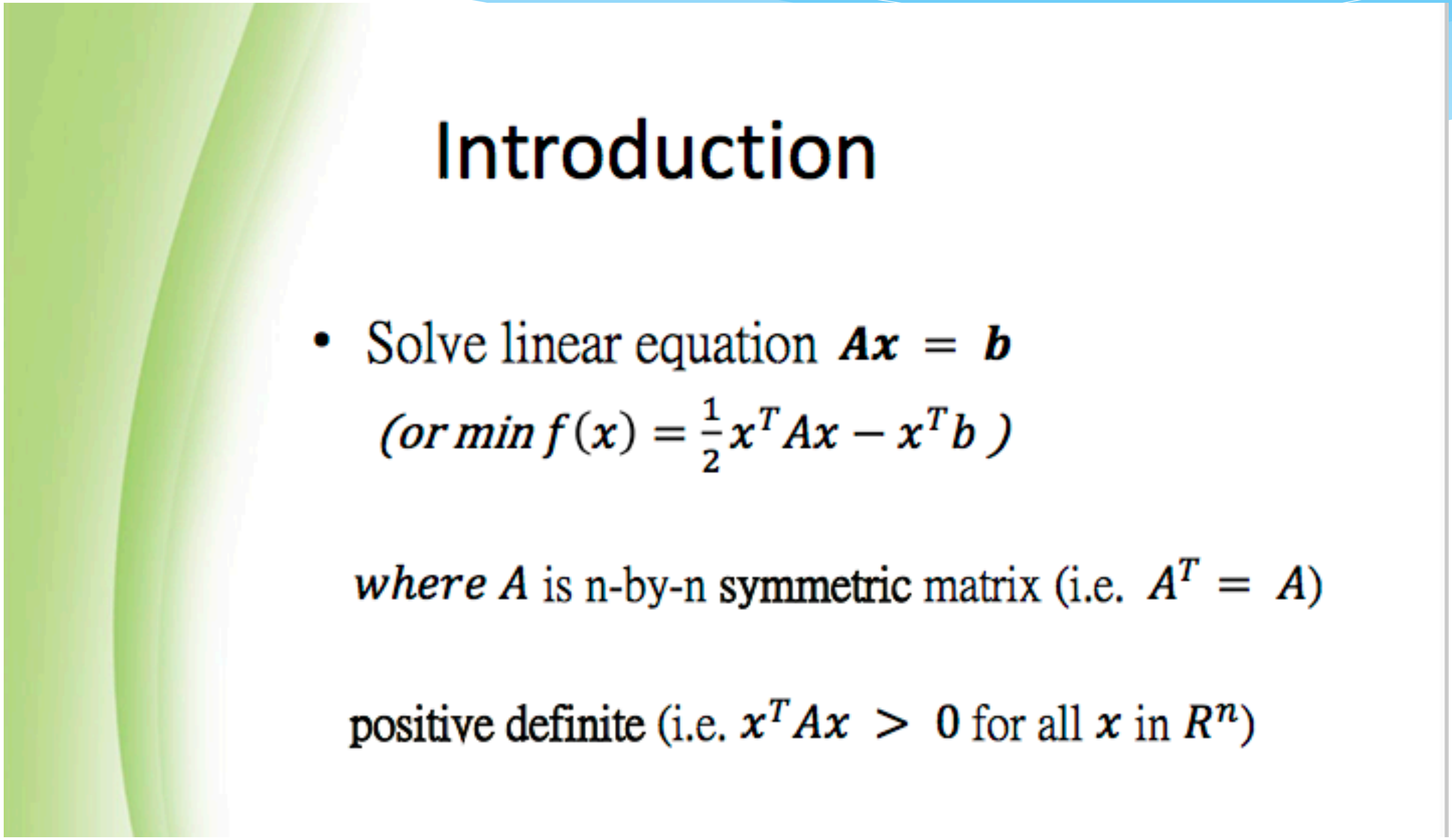
Conjugate Gradient method

Numerical Analysis 2017

AM NDHU



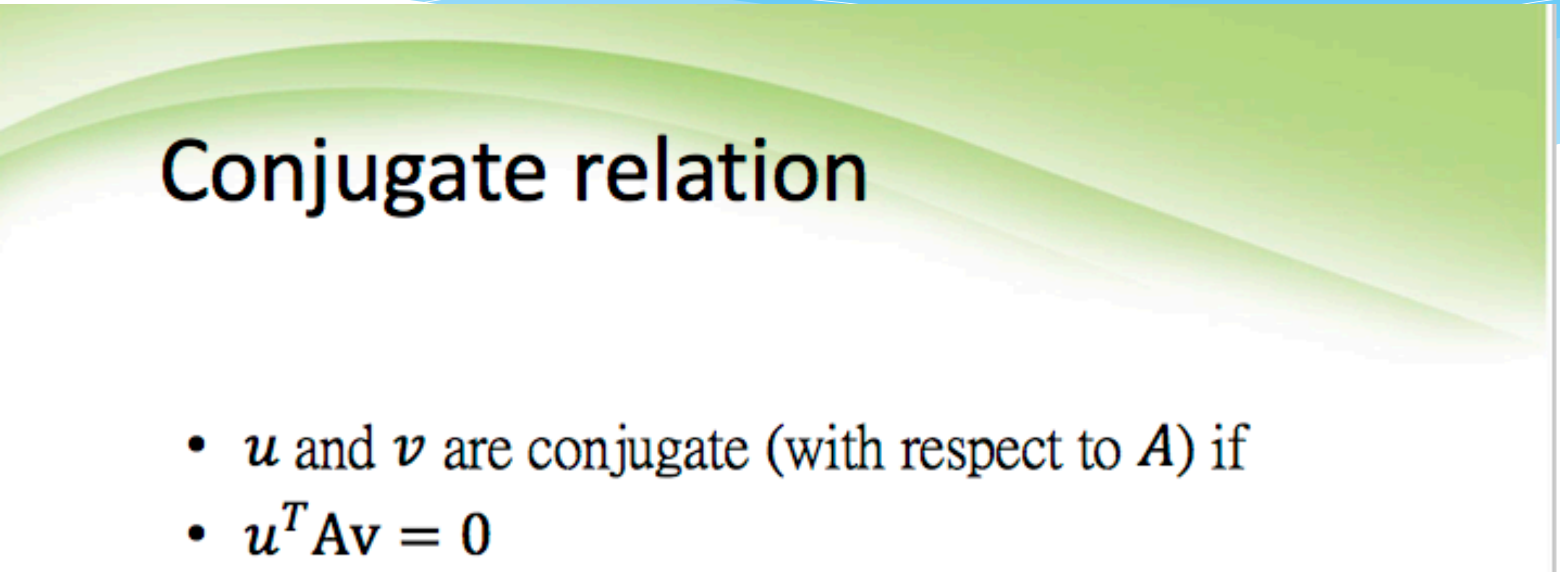
Introduction

- Solve linear equation $\boldsymbol{Ax} = \boldsymbol{b}$

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

  where A is n-by-n symmetric matrix (i.e. $A^T = A$)

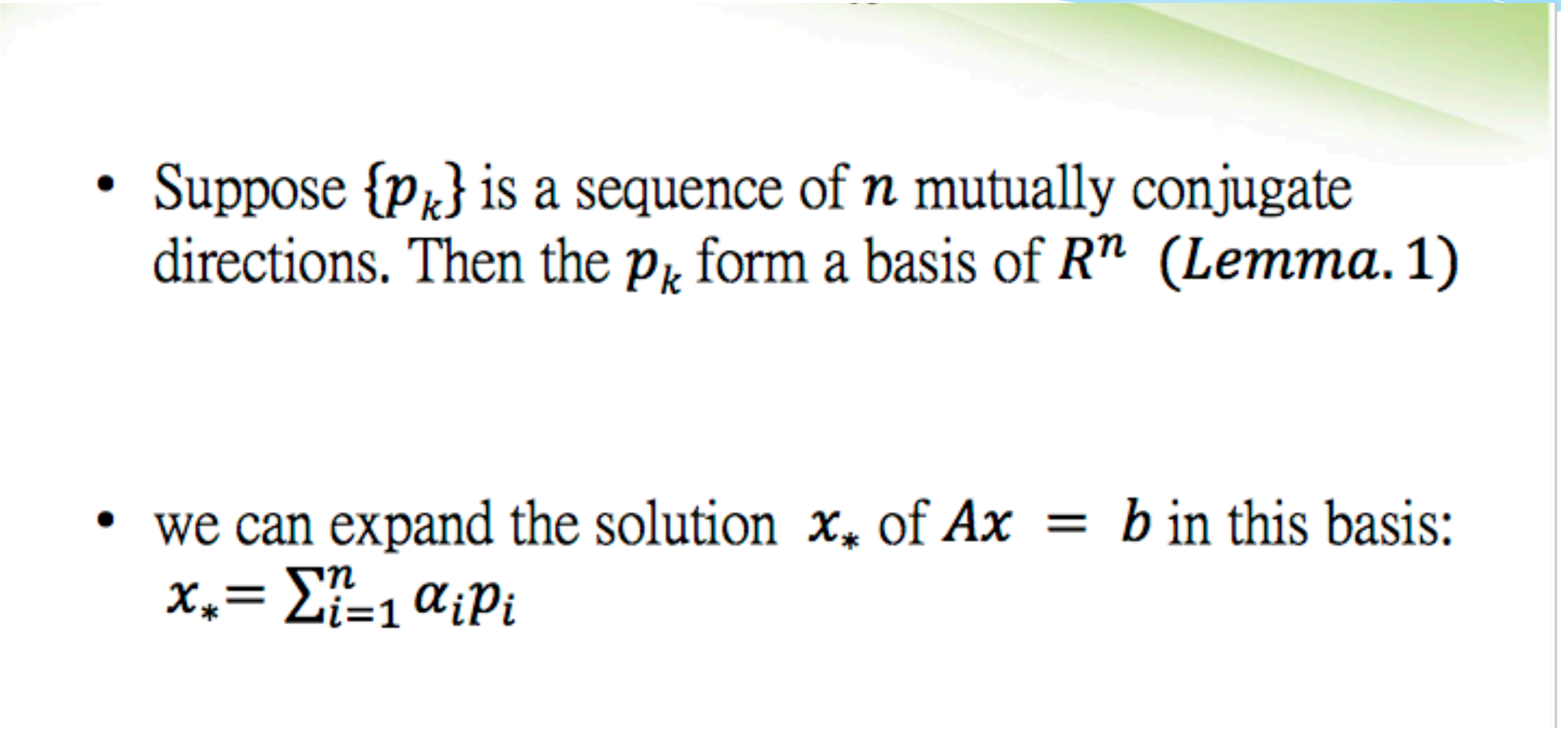
  positive definite (i.e. $x^T Ax > 0$ for all x in R^n)



Conjugate relation

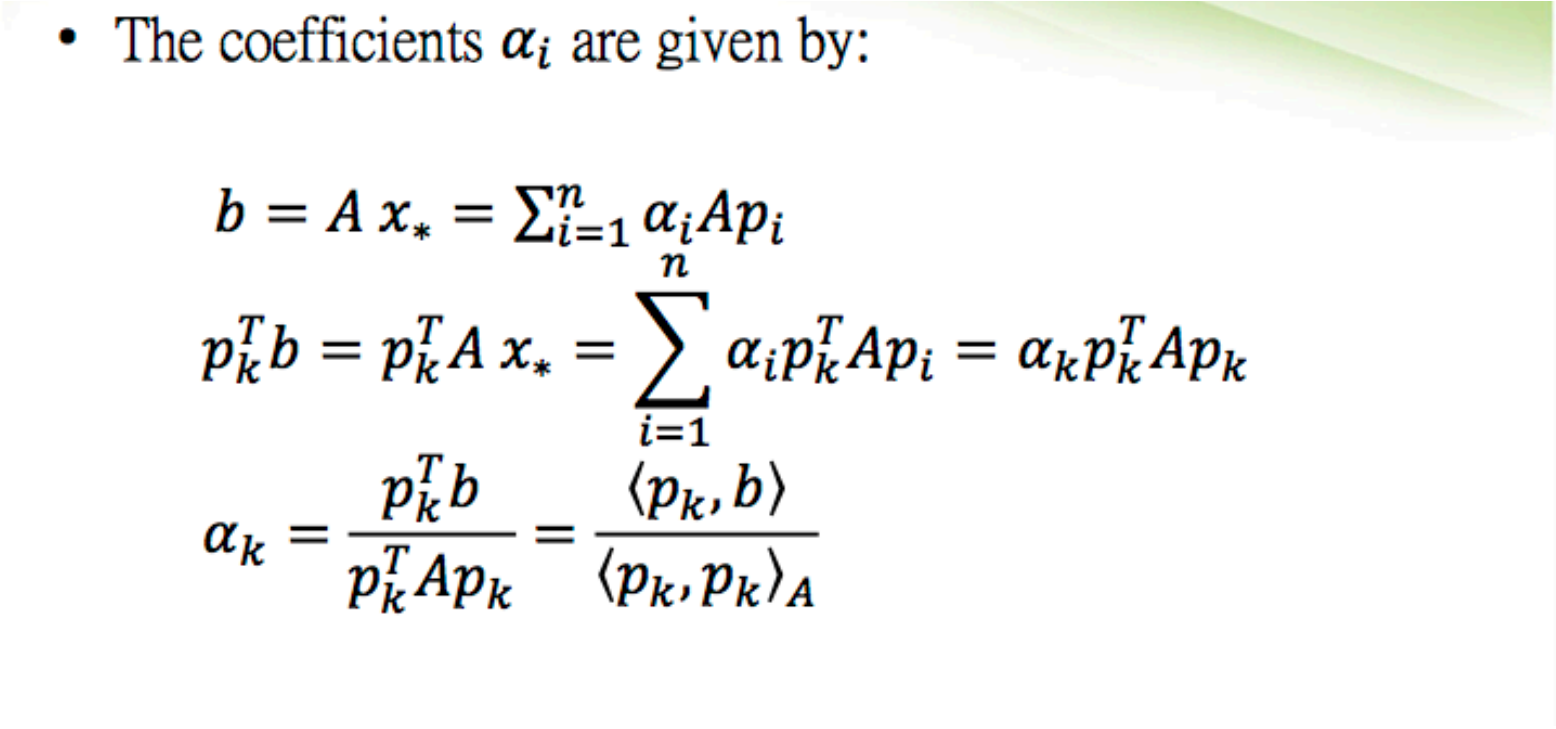
- u and v are conjugate (with respect to A) if
- $u^T \mathrm{Av} = 0$

$$\langle u, v \rangle_A = 0$$



- Suppose $\{p_k\}$ is a sequence of $\boldsymbol{n}$ mutually conjugate directions. Then the $\boldsymbol{p}_k$ form a basis of $\boldsymbol{R}^n$ $(\boldsymbol{Lemma.1})$

- we can expand the solution $\boldsymbol{x}_*$ of $\boldsymbol{Ax} = \boldsymbol{b}$ in this basis: $\boldsymbol{x}_* = \sum_{i=1}^{n} \alpha_i p_i$

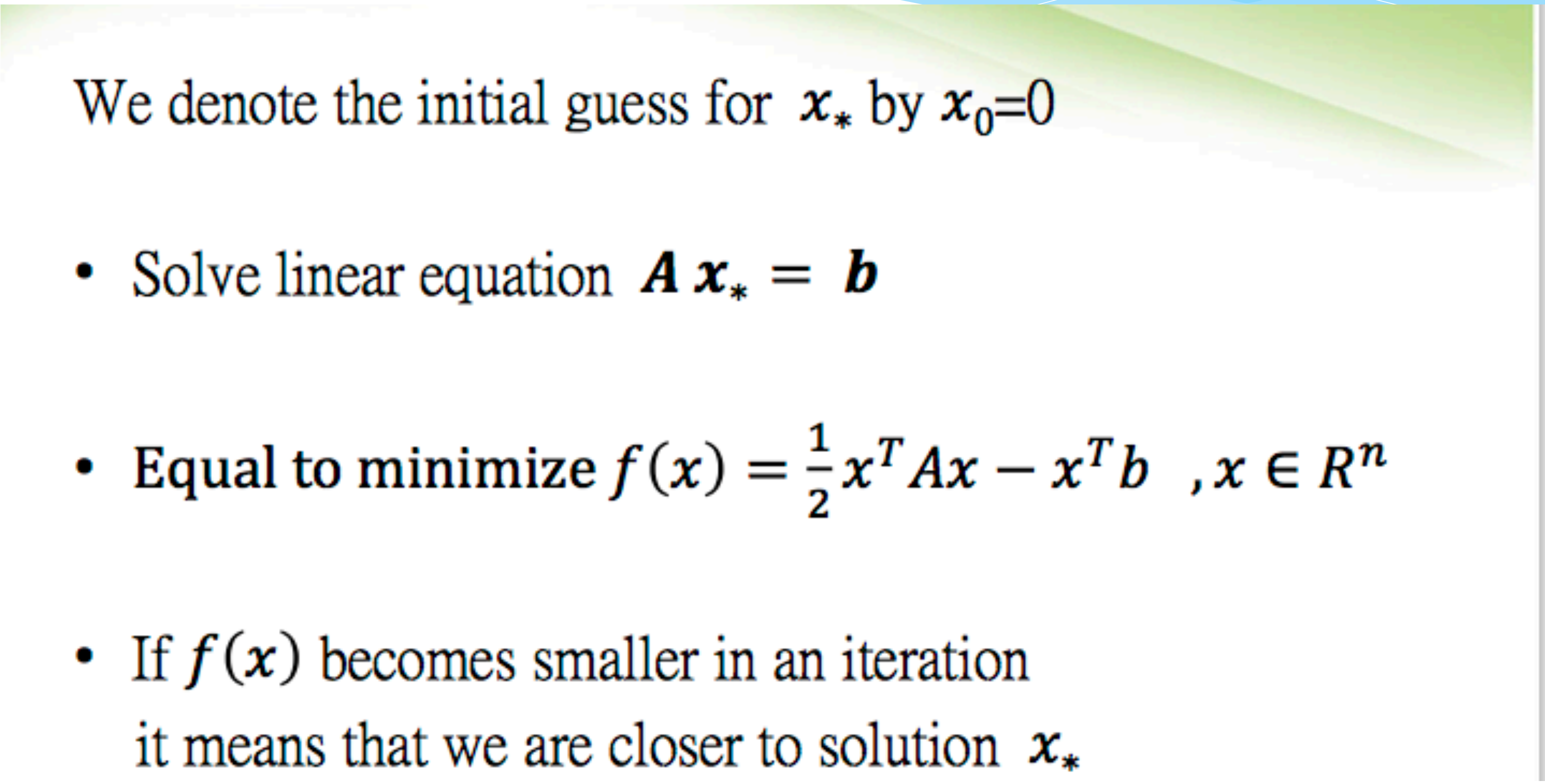


- The coefficients α_i are given by:

$$b = A\,x_* = \textstyle\sum_{i=1}^{n} \alpha_i A p_i$$

$$p_k^T b = p_k^T A\,x_* = \sum_{i=1}^{n} \alpha_i p_k^T A p_i = \alpha_k p_k^T A p_k$$

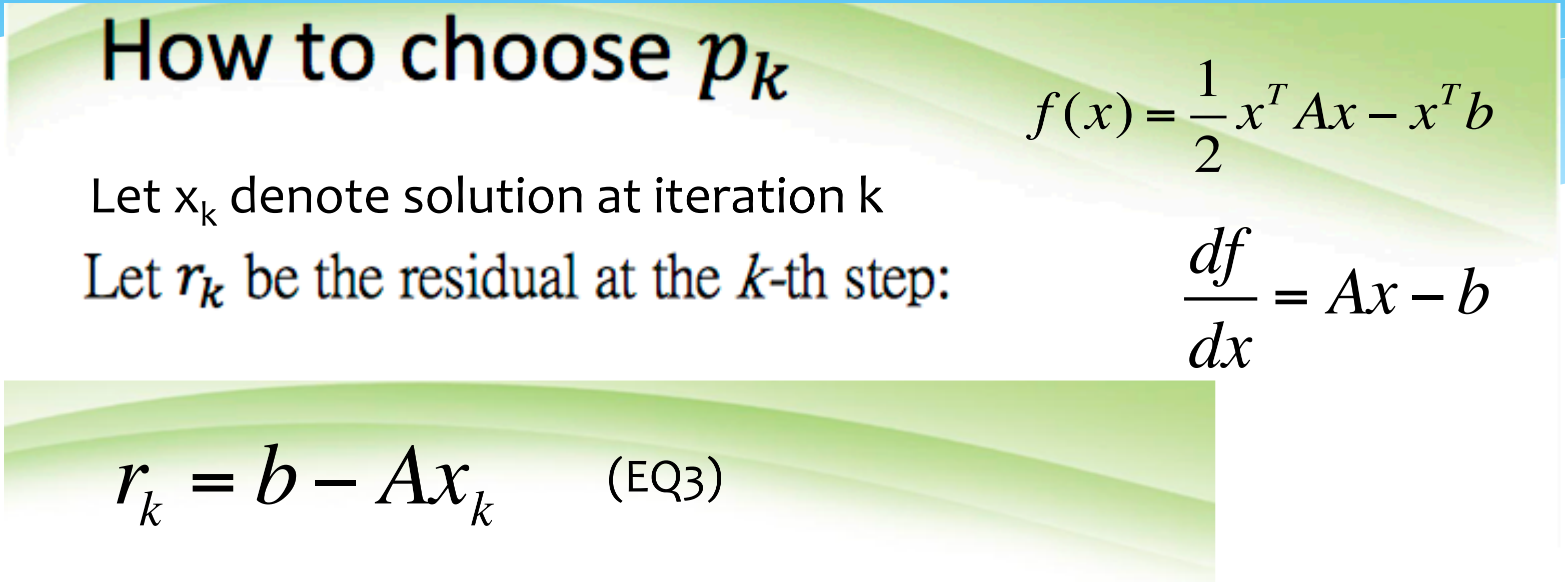
$$\alpha_k = \frac{p_k^T b}{p_k^T A p_k} = \frac{\langle p_k, b\rangle}{\langle p_k, p_k\rangle_A}$$



We denote the initial guess for $\boldsymbol{x}_*$ by $\boldsymbol{x}_0$=0

- Solve linear equation $\boldsymbol{A}\,\boldsymbol{x}_* = \boldsymbol{b}$
- **Equal to minimize** $\boldsymbol{f(x) = \frac{1}{2}x^T A x - x^T b \quad , x \in R^n}$
- If $\boldsymbol{f(x)}$ becomes smaller in an iteration

  it means that we are closer to solution $\boldsymbol{x}_*$



How to choose p_k

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

Let x_k denote solution at iteration k

Let $\boldsymbol{r}_k$ be the residual at the k-th step:

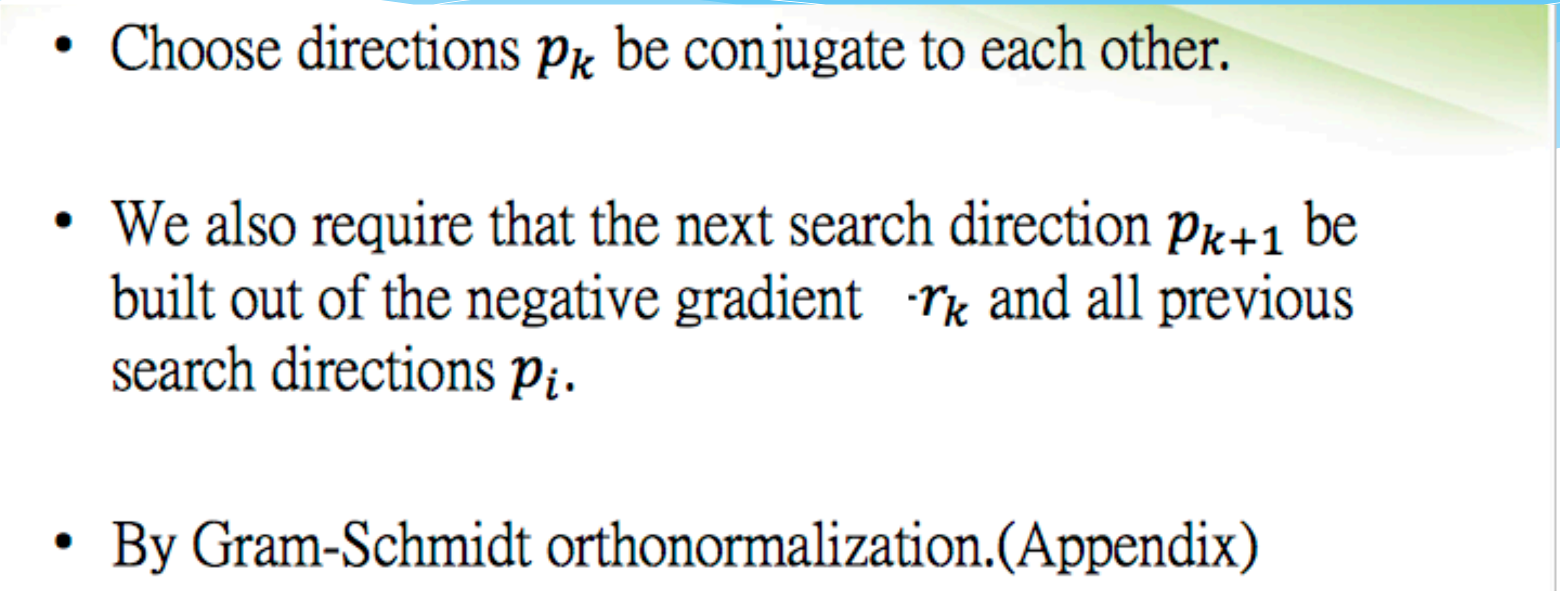
$$\frac{df}{dx} = Ax - b$$

$$r_k = b - Ax_k \qquad \text{(EQ3)}$$

Note: r_k is the negative gradient of f(x) at $x=x_k$

So the gradient descent method would move in direction r_k

the negative gradient of f(x) with respect to x



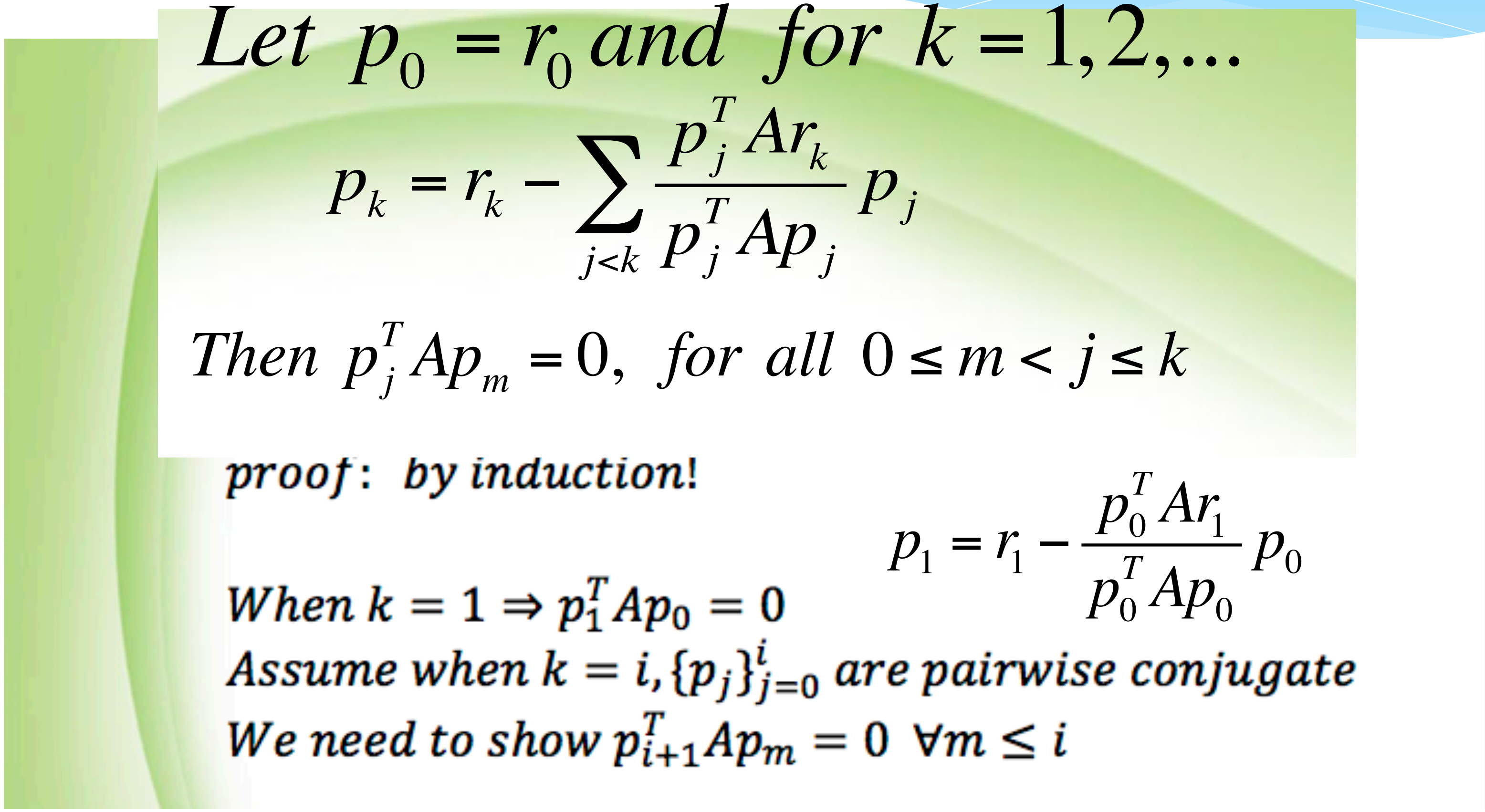
- Choose directions $\boldsymbol{p_k}$ be conjugate to each other.

- We also require that the next search direction $\boldsymbol{p_{k+1}}$ be built out of the negative gradient $-\boldsymbol{r_k}$ and all previous search directions $\boldsymbol{p_i}$.

- By Gram-Schmidt orthonormalization.(Appendix)

$$p_k = r_k - \sum_{i<k} \frac{p_i^T A r_k}{p_i^T A p_i} p_i \, (EQ4)$$

Note: $\{p_k\}$ all pairwise conjugate



$$Let\ \ p_0 = r_0\, and\ \ for\ \ k = 1,2,\ldots$$

$$p_k = r_k - \sum_{j<k} \frac{p_j^T A r_k}{p_j^T A p_j} p_j$$

$$Then\ \ p_j^T A p_m = 0,\ \ for\ \ all\ \ 0 \le m < j \le k$$

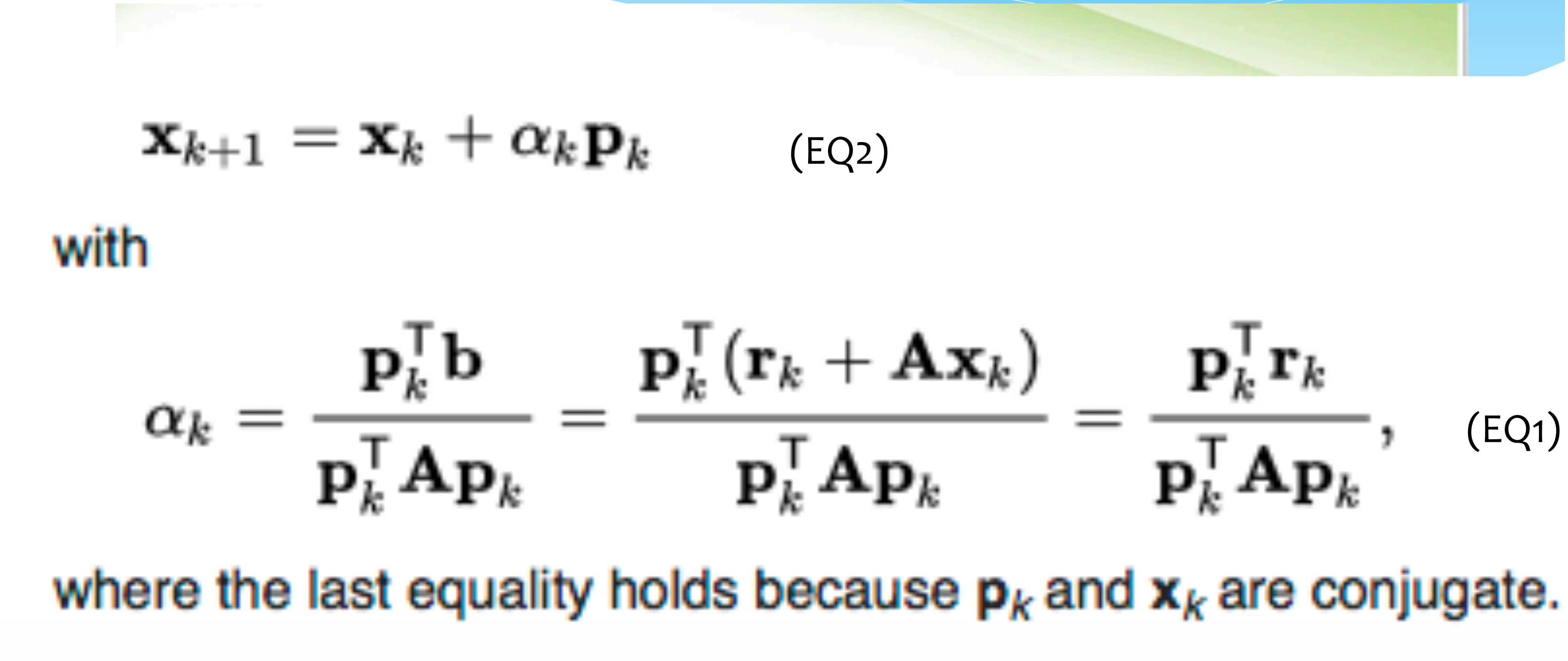
proof: *by induction!*

$$p_1 = r_1 - \frac{p_0^T A r_1}{p_0^T A p_0} p_0$$

When $k = 1 \Rightarrow p_1^T A p_0 = 0$

Assume when $k = i, \{p_j\}_{j=0}^{i}$ *are pairwise conjugate*

We need to show $p_{i+1}^T A p_m = 0\ \ \forall m \le i$



$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$ (EQ2)

with

$$\alpha_k = \frac{\mathbf{p}_k^\mathsf{T} \mathbf{b}}{\mathbf{p}_k^\mathsf{T} \mathbf{A} \mathbf{p}_k} = \frac{\mathbf{p}_k^\mathsf{T} (\mathbf{r}_k + \mathbf{A}\mathbf{x}_k)}{\mathbf{p}_k^\mathsf{T} \mathbf{A} \mathbf{p}_k} = \frac{\mathbf{p}_k^\mathsf{T} \mathbf{r}_k}{\mathbf{p}_k^\mathsf{T} \mathbf{A} \mathbf{p}_k},$$ (EQ1)

where the last equality holds because $\mathbf{p}_k$ and $\mathbf{x}_k$ are conjugate.

$$x_1 = x_0 + a_0 p_0$$

$$x_k = x_{k-1} + a_{k-1} p_{k-1} = x_0 + \sum_{i \le k-1} a_i p_i$$

(EQ. 1) $$\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}$$

(EQ2) $$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

$$r_k = b - Ax_k$$

(EQ3) $$= b - A(x_{k-1} + \alpha_{k-1} p_{k-1})$$

$$= r_{k-1} - \alpha_{k-1} A p_{k-1}$$

(EQ4) $$p_k = r_k - \sum_{j<k} \frac{p_j^T A r_k}{p_j^T A p_j} p_j$$

Note: $\mathbf{p}_k$ is conjugate search direction

Note: r_k is the negative gradient of f(x) at $x=x_k$

(EQ. 1) $$\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}$$

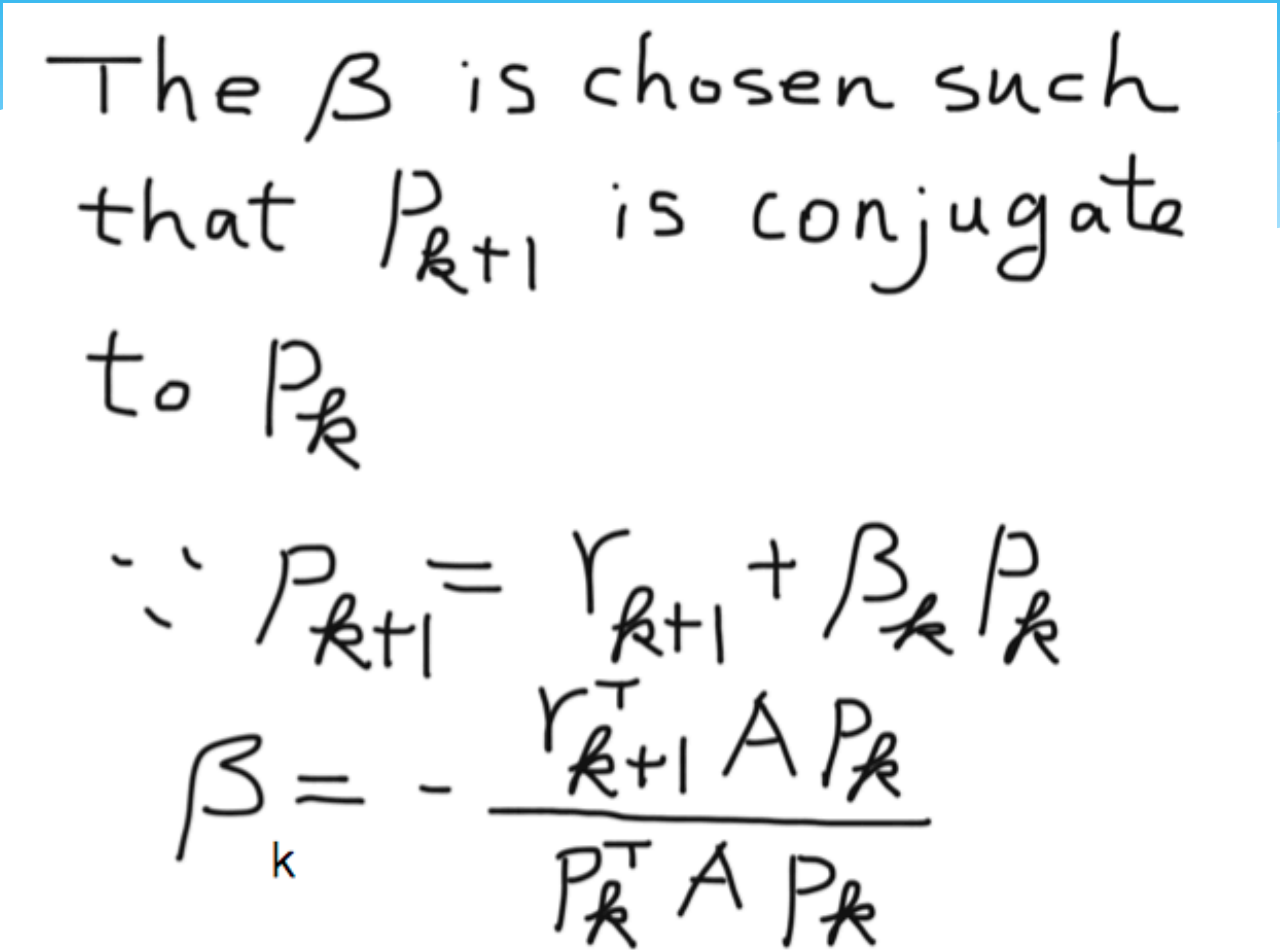
$$\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}$$

r_k : negative gradient

$\because p_k = r_k + \beta_{k-1} p_{k-1}$

r_k and p_{k-1} are conjugate

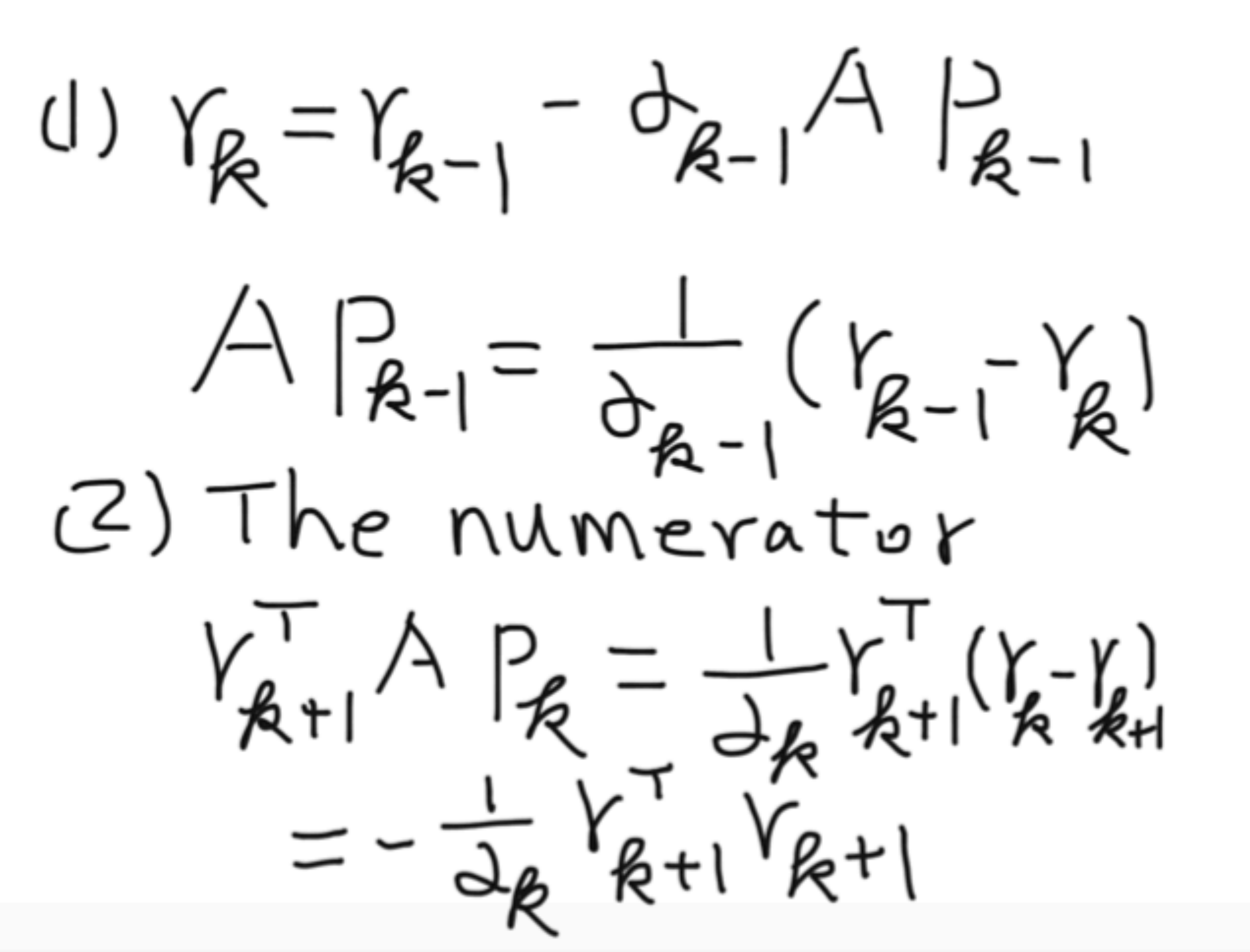
$$\Rightarrow \alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$$



The β is chosen such that p_{k+1} is conjugate to p_k

$$\because\ p_{k+1} = r_{k+1} + \beta_k p_k$$

$$\beta_k = - \frac{r_{k+1}^T A p_k}{p_k^T A p_k}$$



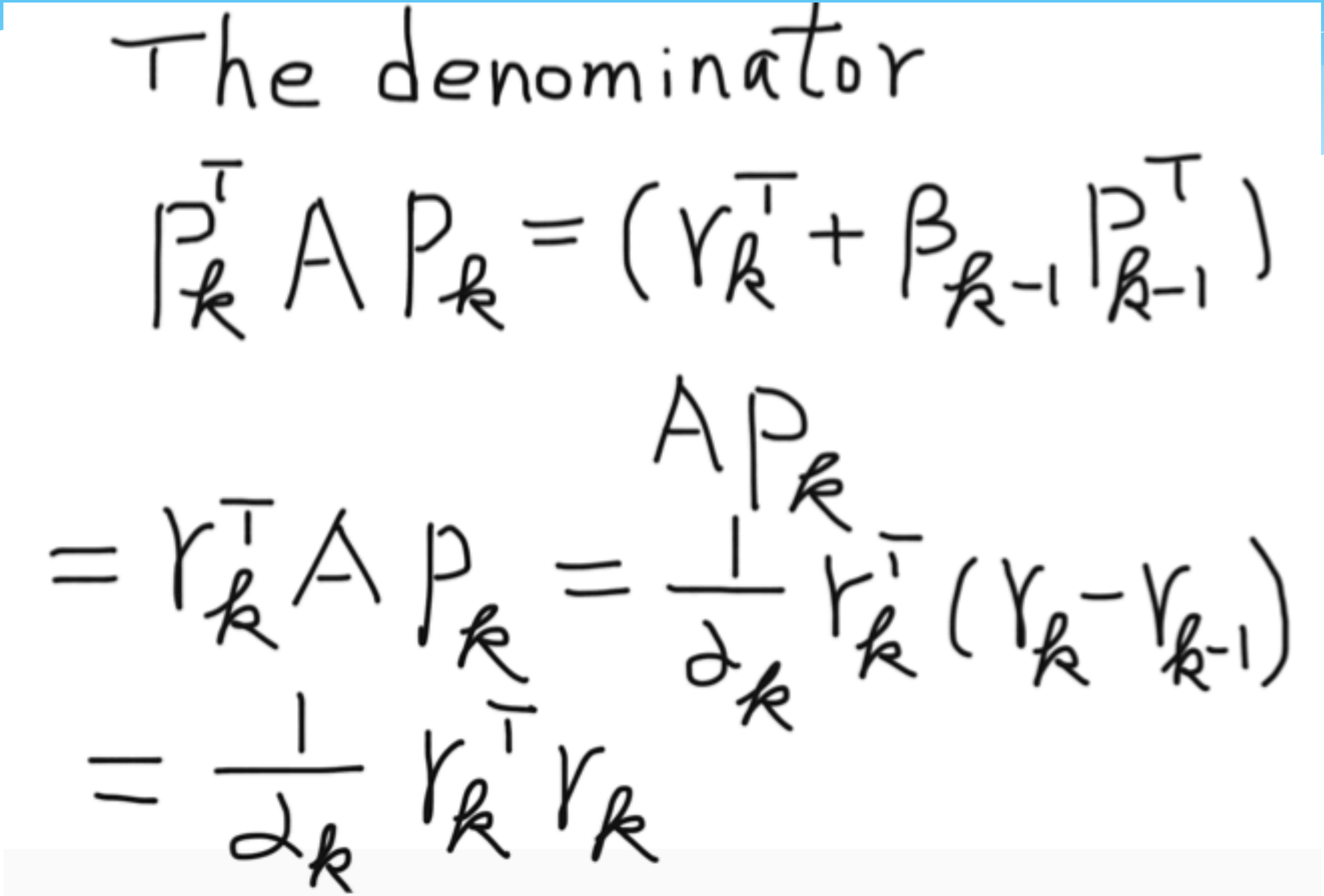
(1) $r_k = r_{k-1} - \alpha_{k-1} A p_{k-1}$

$$A p_{k-1} = \frac{1}{\alpha_{k-1}} (r_{k-1} - r_k)$$

(2) The numerator

$$r_{k+1}^T A p_k = \frac{1}{\alpha_k} r_{k+1}^T (r_k - r_{k+1})$$

$$= -\frac{1}{\alpha_k} r_{k+1}^T r_{k+1}$$



The denominator

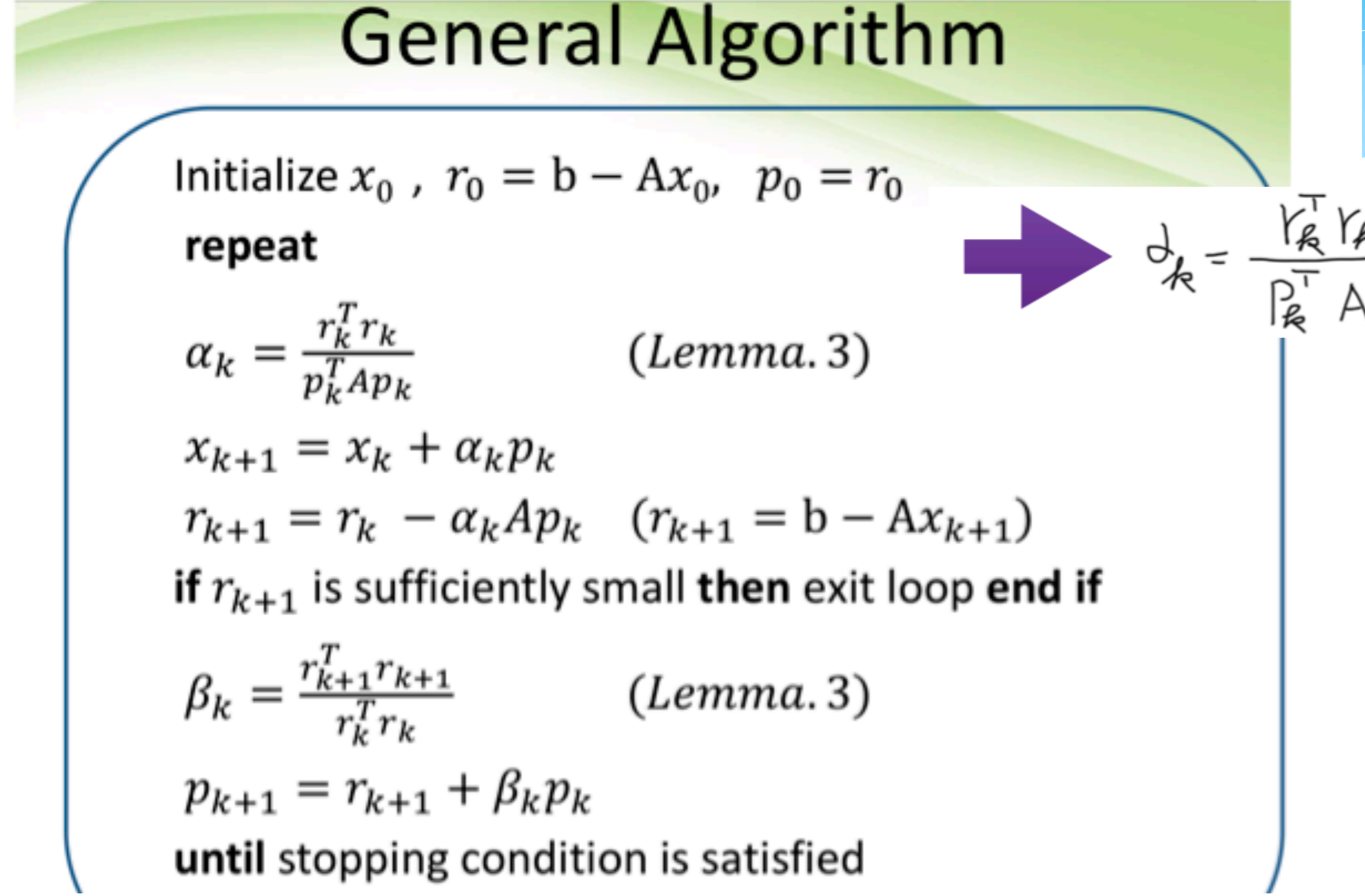
$$p_k^T A p_k = (r_k^T + \beta_{k-1} p_{k-1}^T) \, A p_k$$

$$= r_k^T A p_k = \frac{1}{\alpha_k} r_k^T (r_k - r_{k-1})$$

$$= \frac{1}{\alpha_k} r_k^T r_k$$

$$\beta_k = -\frac{r_{k+1}^T A p_k}{p_k^T A p_k}$$

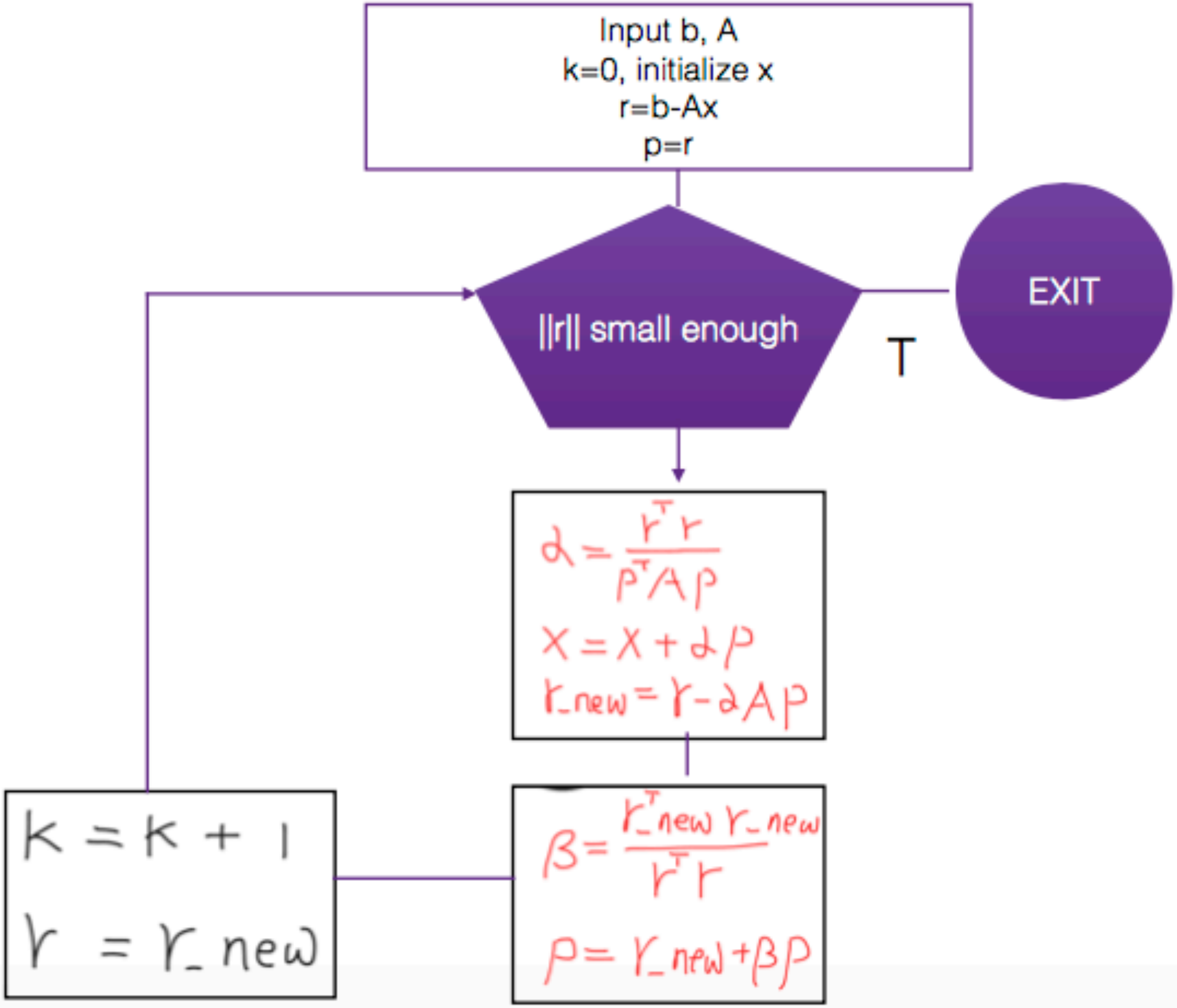
$$\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$



General Algorithm

```
Initialize $x_0$ , $r_0 = \mathrm{b} - \mathrm{A}x_0$,  $p_0 = r_0$
repeat
    $\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$    $(Lemma.3)$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k A p_k$   $(r_{k+1} = \mathrm{b} - \mathrm{A}x_{k+1})$
    if $r_{k+1}$ is sufficiently small then exit loop end if
    $\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$    $(Lemma.3)$
    $p_{k+1} = r_{k+1} + \beta_k p_k$
until stopping condition is satisfied
```

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



Input b, A
k=0, initialize x
r=b-Ax
p=r

||r|| small enough

T

EXIT

$\alpha = \frac{r^T r}{p^T A p}$

$x = x + \alpha p$

$r\_new = r - \alpha A p$

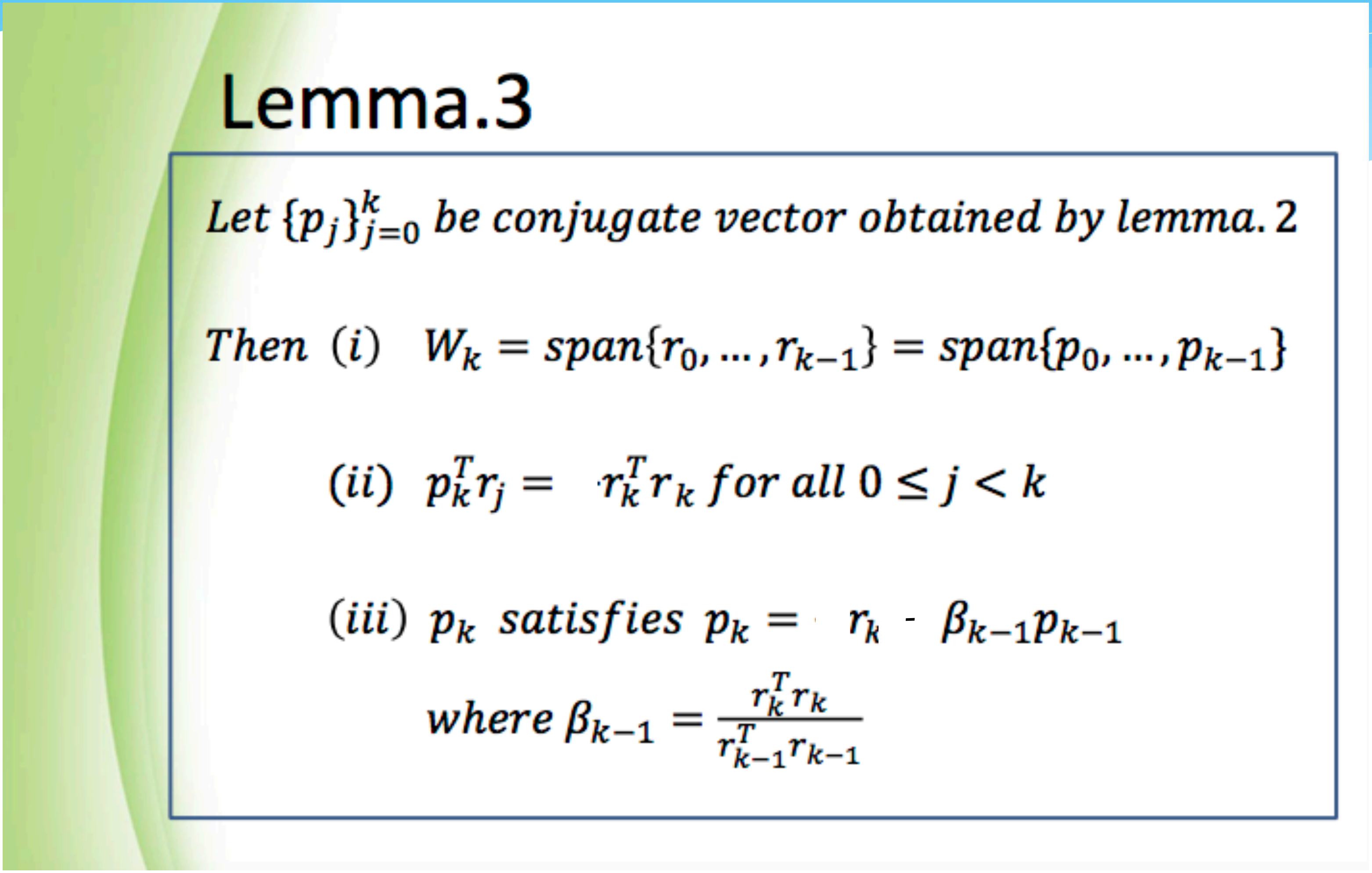
$\beta = \frac{r\_new^T r\_new}{r^T r}$

$p = r\_new + \beta p$

$k = k + 1$

$r = r\_new$

```
function x=CG_lin(A,b)
k=0;
N=size(A,2);
x=rand(N,1)*2-1;
r=b-A*x;
p=r;
hc= norm(r) < 10^-6 ;
while ~hc
    a=(r'*r)/(p'*A*p);
    x=x+a*p;
    r_new=r-a*A*p;
    beta=(r_new'*r_new)/(r'*r);
    p=r_new+beta*p;
    r=r_new;
    hc= norm(r) < 10^-6 ;
    k=k+1;
    F(k)=1/2*x'*A*x-x'*b;
    if mod(k,10)==0
        fprintf('iter = %d, error = %f, f(x)=%f\n',k,norm(r),F(k))
    end
    Err(k)=norm(r);
end
```



Lemma.3

Let $\{p_j\}_{j=0}^{k}$ *be conjugate vector obtained by lemma.* 2

Then (i) $W_k = span\{r_0, \dots, r_{k-1}\} = span\{p_0, \dots, p_{k-1}\}$

(ii) $p_k^T r_j = r_k^T r_k$ *for all* $0 \le j < k$

(iii) p_k *satisfies* $p_k = r_k - \beta_{k-1} p_{k-1}$

where $\beta_{k-1} = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$

$proof:$

(i) $follows\ directly\ from\ lemma.2$

$(by\ induction\ p_0 = \ \ r_0)$

(ii) $for\ j = k,$

$by\ (i), \langle r_k, p_l \rangle = 0\ for\ l < k$

$p_k^T r_k = \ \ r_k^T r_k\ by\ lemma.2$

$for\ j < k$

$(x_k - x_j) \in W_k \Rightarrow p_k^T A(x_k - x_j) = 0$

$therefore,$

$p_k^T(r_k - r_j) = p_k^T A(x_k - x_j) = 0$

$\Rightarrow p_k^T r_j = \ \ r_k^T r_k\ by\ lemma.\ 2$

(iii) *write* $p_k \in W_{k+1}$ *as linear combination of* $\{r_j\}_{j=0}^{k}$

(which form an orthogonal basis,)

$$p_k = \sum_{j=0}^{k} \frac{p_k^T r_j}{r_j^T r_j} r_j = \sum_{j=0}^{k} \frac{r_k^T r_k}{r_j^T r_j} r_j$$

$$= r_k - \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}} \sum_{j=0}^{k-1} \frac{r_{k-1}^T r_{k-1}}{r_j^T r_j} r_j$$

$$= r_k - \beta_{k-1} \sum_{j=0}^{k-1} \frac{p_{k-1}^T r_j}{r_j^T r_j} r_j$$

$$= r_k - \beta_{k-1} p_{k-1}$$

$$where\ \beta_{k-1} = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$$

Exercise 1 (2017/10)

Problem 1. Let B denote an m-by-n matrix and c denote an n-by-1 vector, where m > n. Bz = c is a linear system, where z collects n unknowns Since B is not square, it is unable to directly solve it by the matlab statement, inv. It is expected to solve it by the conjugate gradient descent method.

a. State why and how to Bz=c by the conjugate gradient descent method.
b. Please design a numerical experiment to generate B and c for some m > n, such as m=1000 and n=500. Try to solve it by function CG_lin

Problem 2. The mnist database of handwritten digits has a training set of 60,000 examples, and a testing set of 10,000 examples. The digits have been size-normalized ad centered in a fixed-size image.

http://yann.lecun.com/exdb/mnist/

a. Discuss applicability of the conjugate gradient descent method for pattern recognition of handwritten digits.
b. Design a numerical experiment to apply matlab function GC_lin to pattern recognition of handwritten digits. Verify effectiveness of GC_lin by this database.