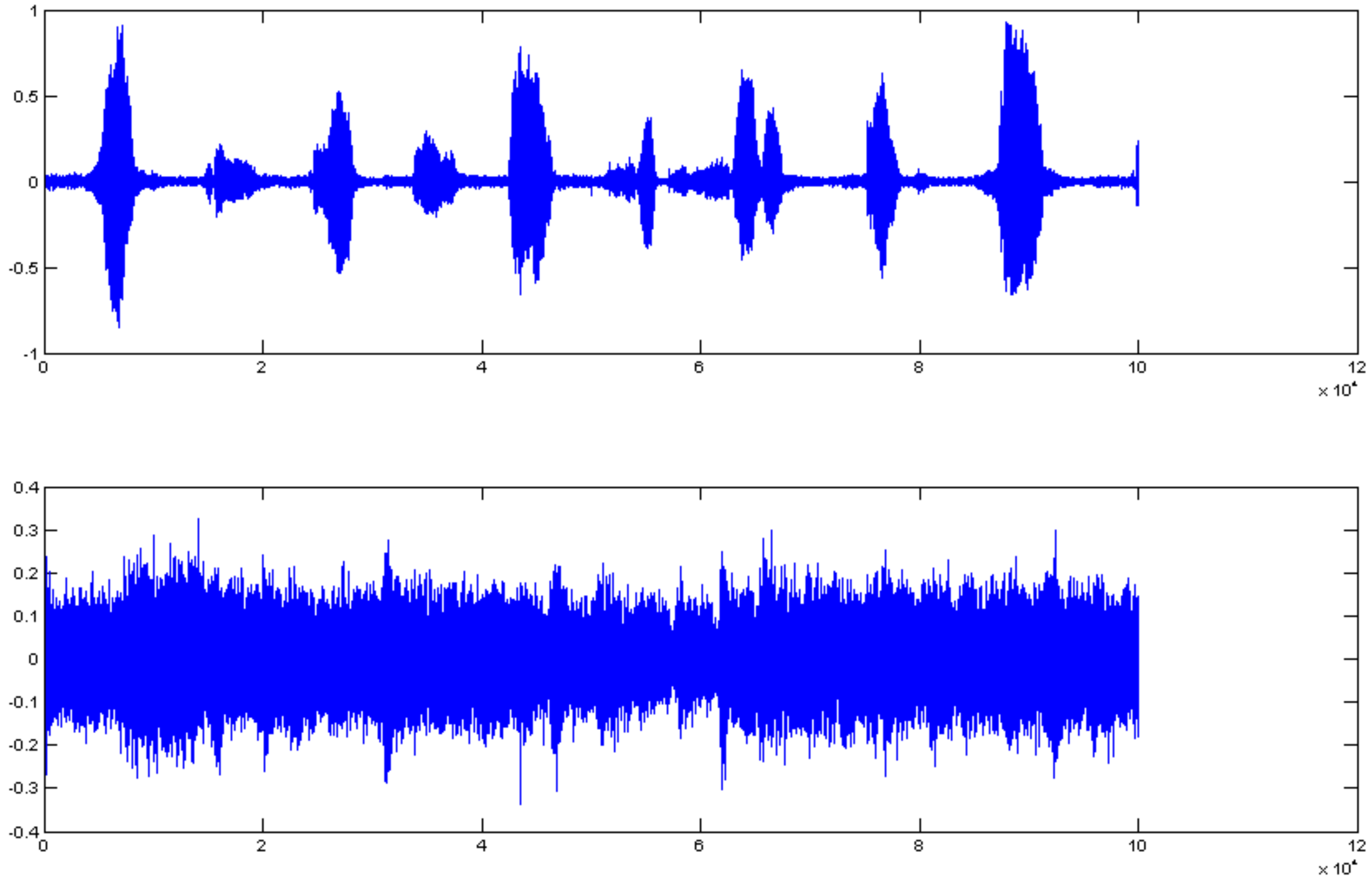
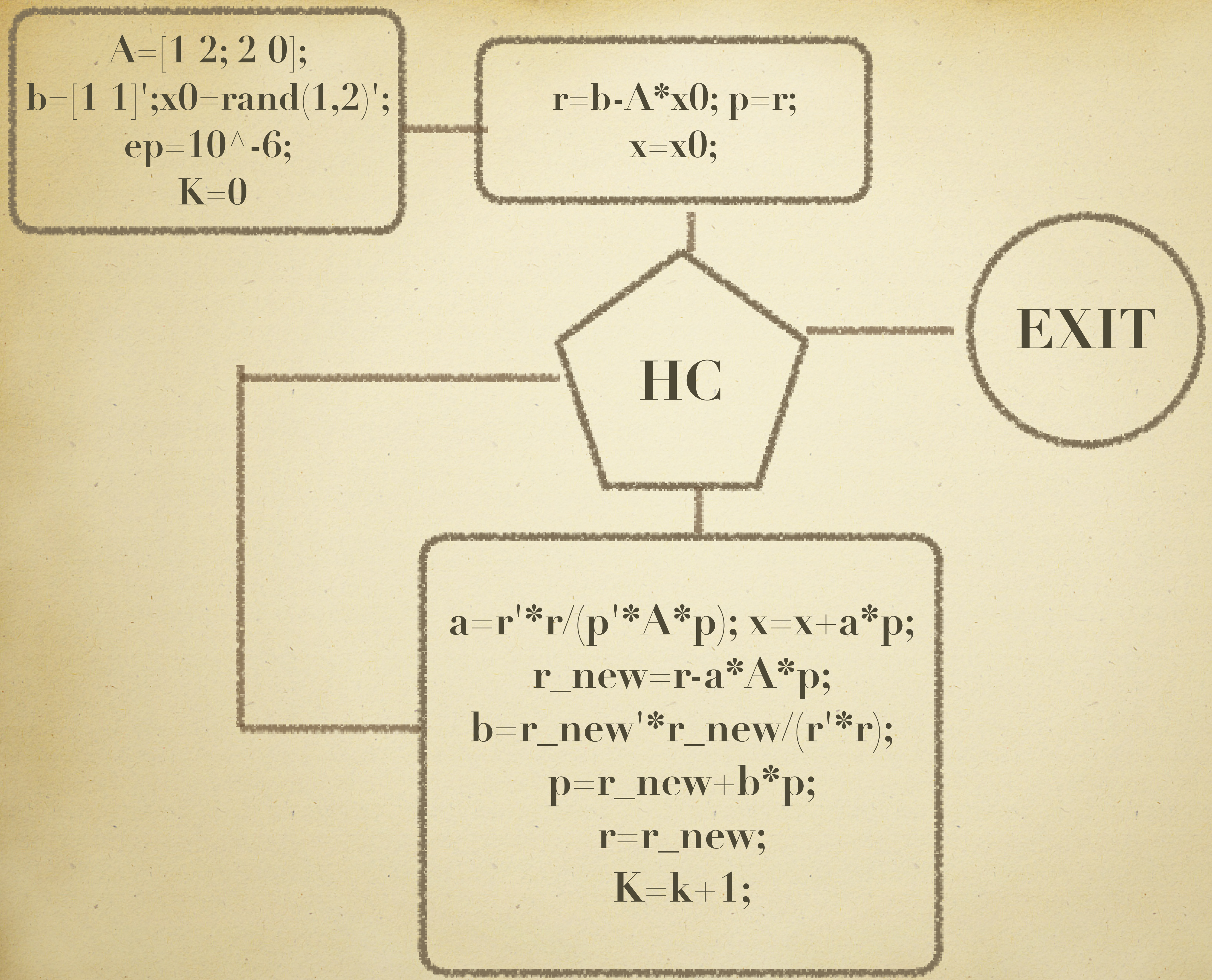
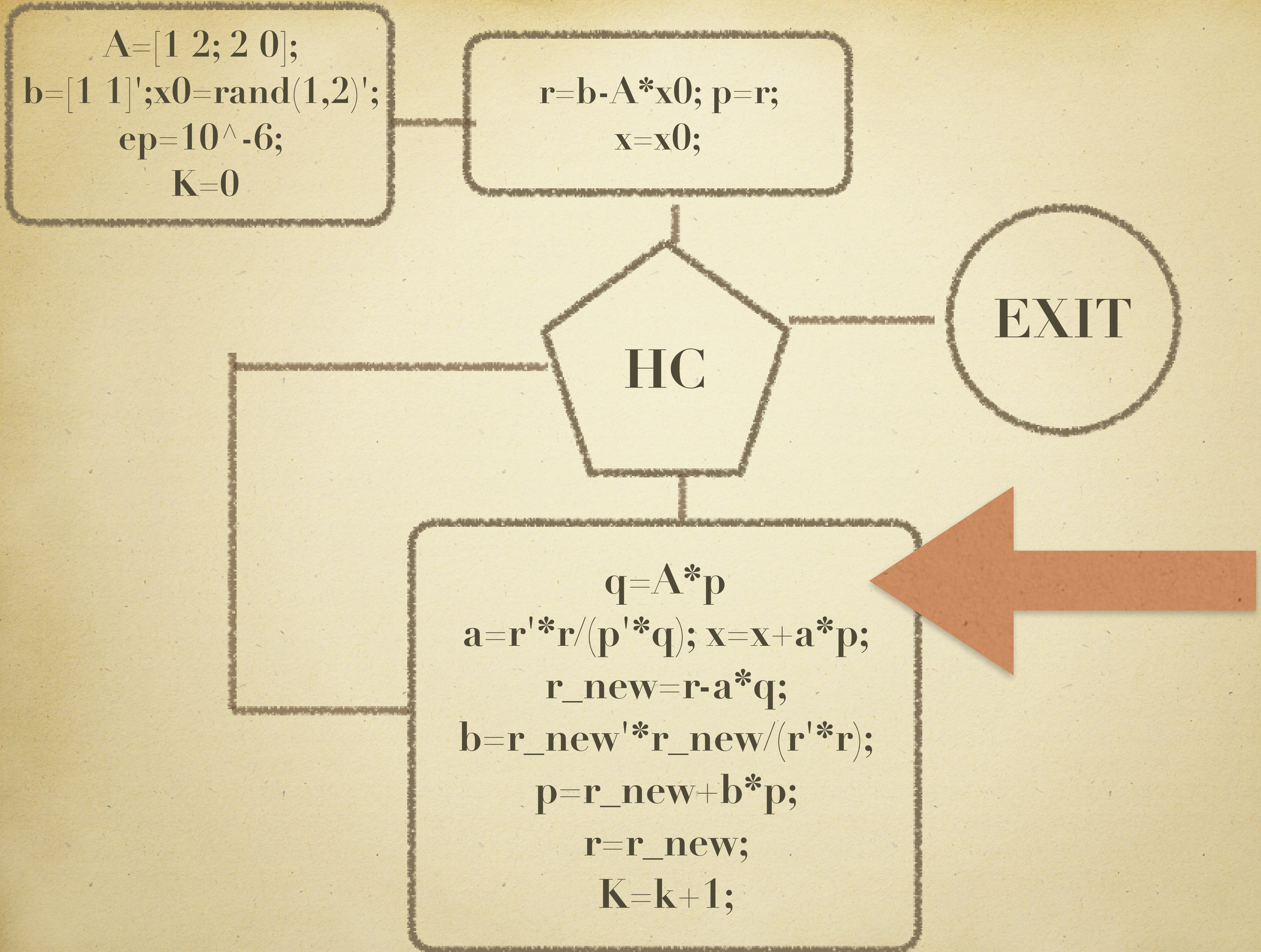


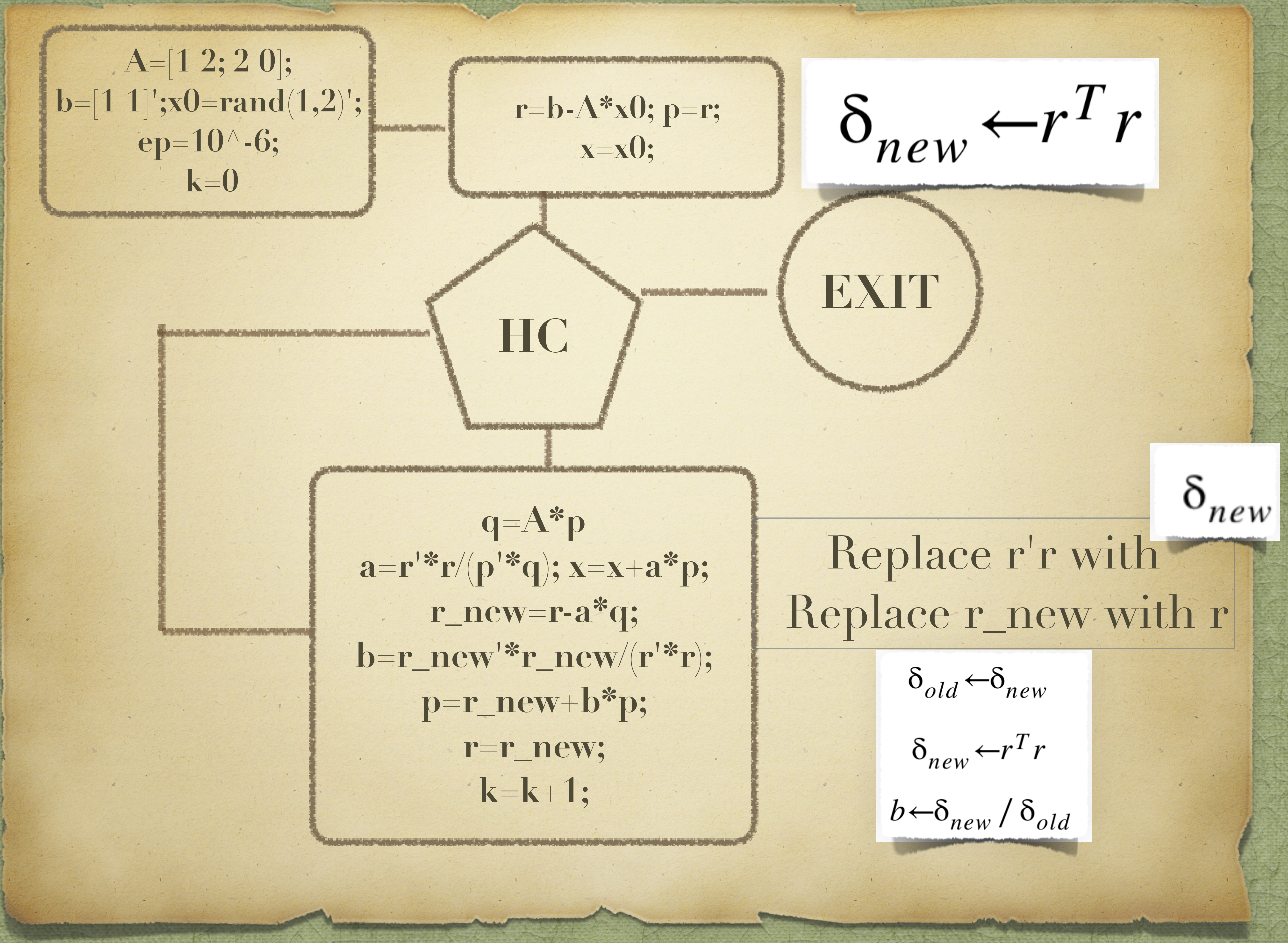
Nonlinear conjugate gradient method

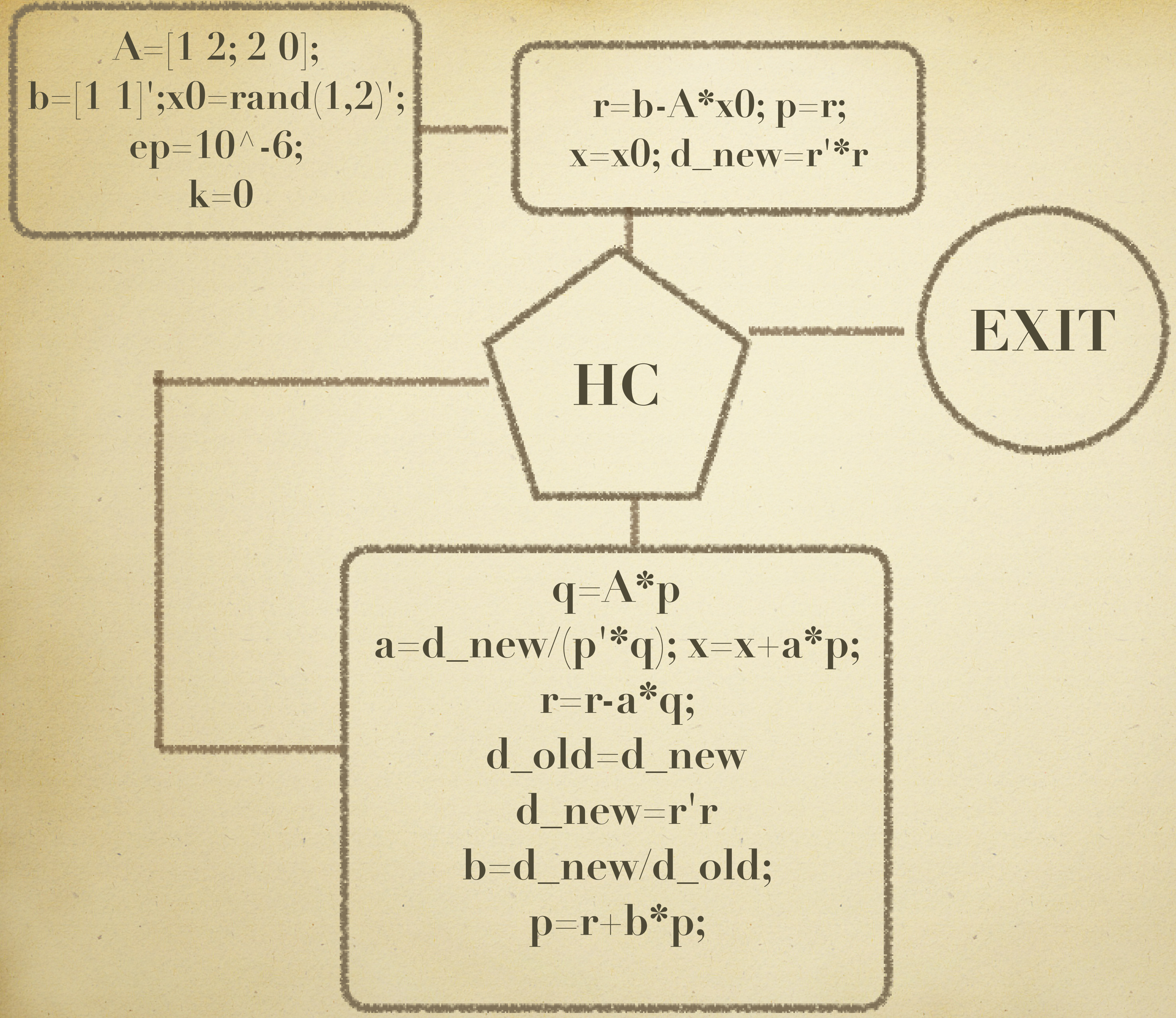
Outline

- Linear conjugate gradient method
- Nonlinear conjugate gradient method
- Line search by Bracketing
- Brent's method


A=[1 2; 2 0];
b=[1 1]';x0=rand(1,2)';
ep=10^-6;
K=0
r=b-A*x0; p=r;
x=x0;
HC
EXIT
a=r'*r/(p'*A*p); x=x+a*p;
r_new=r-a*A*p;
b=r_new'*r_new/(r'*r);
p=r_new+b*p;
r=r_new;
K=k+1;


A=[1 2; 2 0];
b=[1 1]';x0=rand(1,2)';
ep=10^-6;
K=0
r=b-A*x0; p=r;
x=x0;
HC
EXIT
q=A*p
a=r'*r/(p'*q); x=x+a*p;
r_new=r-a*q;
b=r_new'*r_new/(r'*r);
p=r_new+b*p;
r=r_new;
K=k+1;

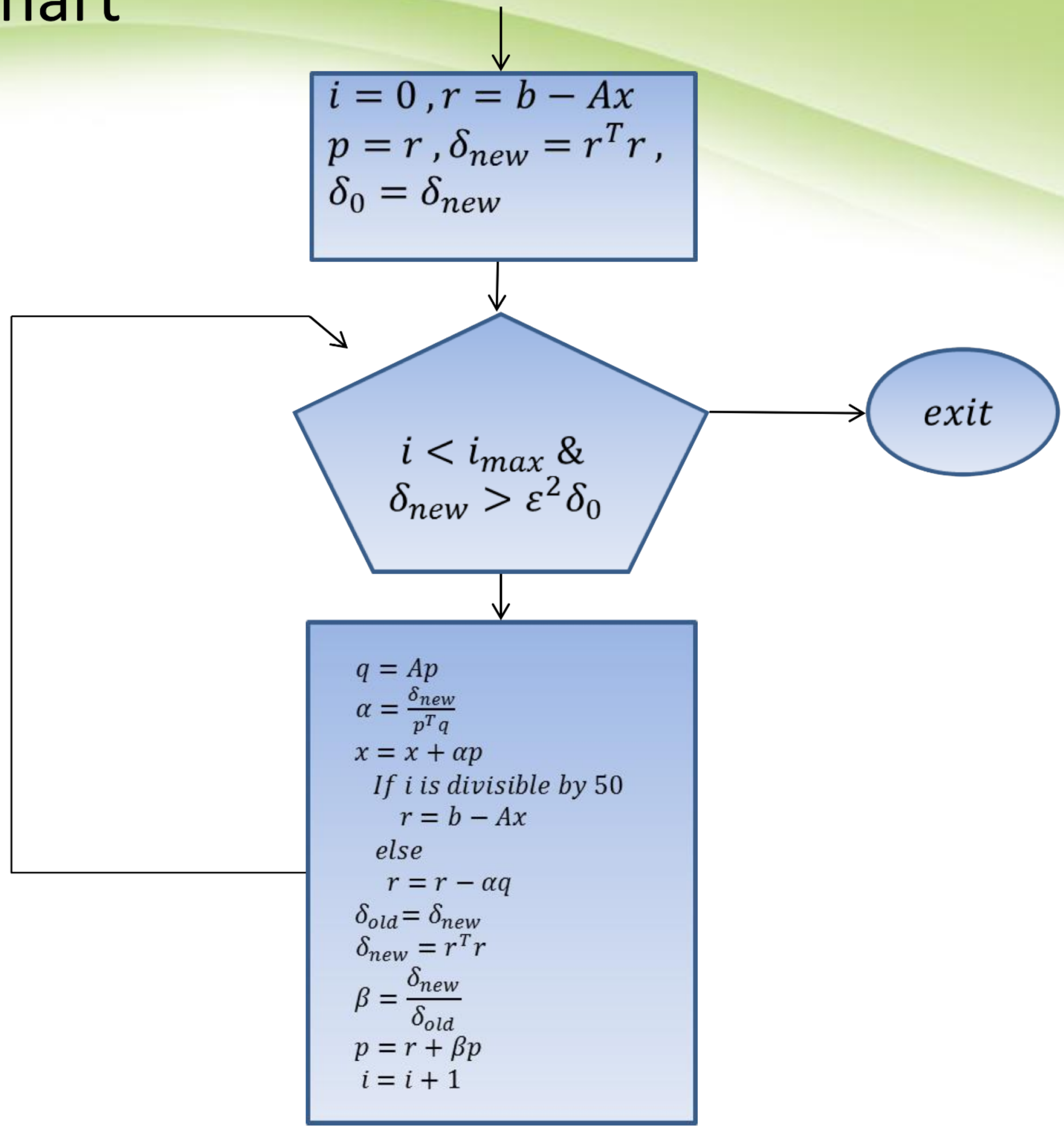

A=[1 2; 2 0];
b=[1 1]';x0=rand(1,2)';
ep=10^-6;
k=0
r=b-A*x0; p=r;
x=x0;
$\delta_{new} \leftarrow r^T r$
HC
EXIT
q=A*p
a=r'*r/(p'*q); x=x+a*p;
r_new=r-a*q;
b=r_new'*r_new/(r'*r);
p=r_new+b*p;
r=r_new;
k=k+1;
δ_{new}
Replace r'r with
Replace r_new with r
$\delta_{old} \leftarrow \delta_{new}$
$\delta_{new} \leftarrow r^T r$
$b \leftarrow \delta_{new} / \delta_{old}$


A=[1 2; 2 0];
b=[1 1]';x0=rand(1,2)';
ep=10^-6;
k=0
r=b-A*x0; p=r;
x=x0; d_new=r'*r
HC
EXIT
q=A*p
a=d_new/(p'*q); x=x+a*p;
r=r-a*q;
d_old=d_new
d_new=r'r
b=d_new/d_old;
p=r+b*p;

Matlab Toolbox

- Toolbox[1]
- Paper[2]
- Paper[3]

Flow chart


$i = 0, r = b - Ax$
$p = r, \delta_{new} = r^T r,$
$\delta_0 = \delta_{new}$
$i < i_{max}$ &
$\delta_{new} > \varepsilon^2 \delta_0$
exit
$q = Ap$
$\alpha = \frac{\delta_{new}}{p^T q}$
$x = x + \alpha p$
If i is divisible by 50
$r = b - Ax$
else
$r = r - \alpha q$
$\delta_{old} = \delta_{new}$
$\delta_{new} = r^T r$
$\beta = \frac{\delta_{new}}{\delta_{old}}$
$p = r + \beta p$
$i = i + 1$

Nonlinear Conjugate Gradient

$i = 0\ , k = 0 , r = -f'(x) , p = r , \delta_{new} = r^T r\ ,\ \delta_0 = \delta_{new}$

$While\ i < i_{max}\ and\ \delta_{new} > \varepsilon^2 \delta_0 \quad do$

$j = 0$

$\delta_p = p^T p$

Do

$$\alpha = \frac{-[f'(x)]^T p}{p^T f''(x) p}$$

$x = x + \alpha p$

$j = j + 1$

$while\ j < j_{max}\ and\ \alpha^2 \delta_p > \varepsilon^2$

$r = -f'(x)$

$\delta_{old} = \delta_{new}$

$\delta_{new} = r^T r$

$$\beta = \frac{\delta_{new}}{\delta_{old}}$$

$p = r + \beta p$

$k = k + 1$

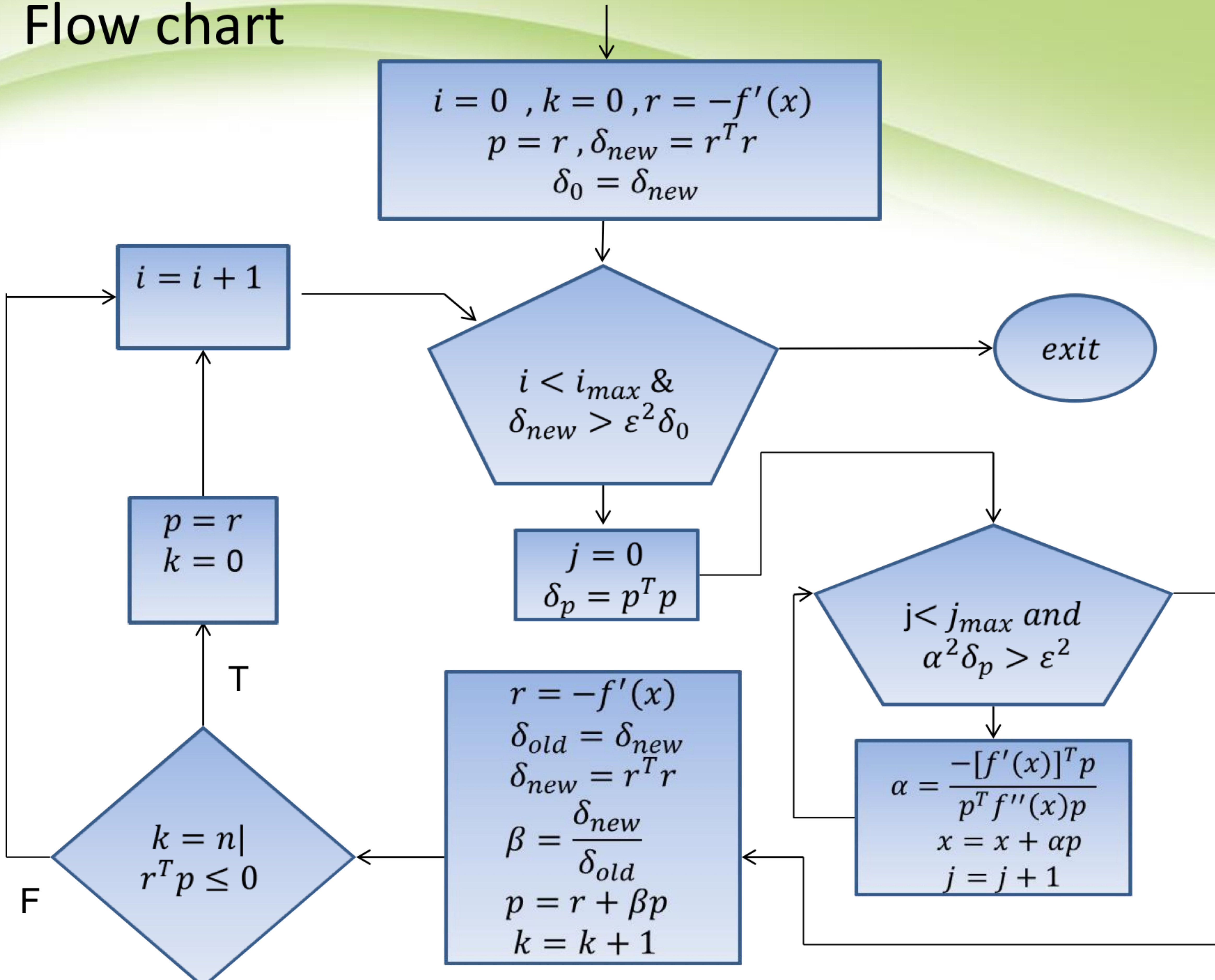
$If\ k = n\ or\ r^T p \le 0$

$p = r$

$k = 0$

$i = i + 1$

Flow chart



$i = 0\ , k = 0 , r = -f'(x)$
$p = r , \delta_{new} = r^T r$
$\delta_0 = \delta_{new}$
$i = i + 1$
$i < i_{max}$ &
$\delta_{new} > \varepsilon^2 \delta_0$
exit
$p = r$
$k = 0$
$j = 0$
$\delta_p = p^T p$
j< j_{max} and
$\alpha^2 \delta_p > \varepsilon^2$
T
$r = -f'(x)$
$\delta_{old} = \delta_{new}$
$\delta_{new} = r^T r$
$\beta = \frac{\delta_{new}}{\delta_{old}}$
$p = r + \beta p$
$k = k + 1$
$\alpha = \frac{-[f'(x)]^T p}{p^T f''(x) p}$
$x = x + \alpha p$
$j = j + 1$
$k = n|$
$r^T p \leq 0$
F

Algorithm RBF-Net(K, λ, O)

Input:

Sequence of labeled training patterns $\mathbf{Z} = \langle(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_l, y_l)\rangle$

Number of RBF centers K

Regularization constant λ

Number of iterations O

Initialize:

Run K-means clustering to find initial values for μ_k and determine $\sigma_k, k = 1, \ldots, K$, as the distance between μ_k and the closest μ_i $(i \neq k)$.

Do for $o = 1 : O$,

1. Compute optimal output weights $\mathbf{w} = \left(G^\top G + 2\frac{\lambda}{l}\mathbf{I}\right)^{-1} G^\top \mathbf{y}$

2a. Compute gradients $\frac{\partial}{\partial \mu_k} E$ and $\frac{\partial}{\partial \sigma_k} E$ as in (28) and (27) with optimal $\mathbf{w}$ and form a gradient vector $\mathbf{v}$

2b. Estimate the conjugate direction $\overline{\mathbf{v}}$ with Fletcher-Reeves-Polak-Ribiere CG-Method (Press et al., 1992)

3a. Perform a line search to find the minimizing step size δ in direction $\overline{\mathbf{v}}$; in each evaluation of E recompute the optimal output weights $\mathbf{w}$ as in line 1

3b. Update μ_k and σ_k with $\overline{\mathbf{v}}$ and δ

Output: Optimized RBF net

Line search

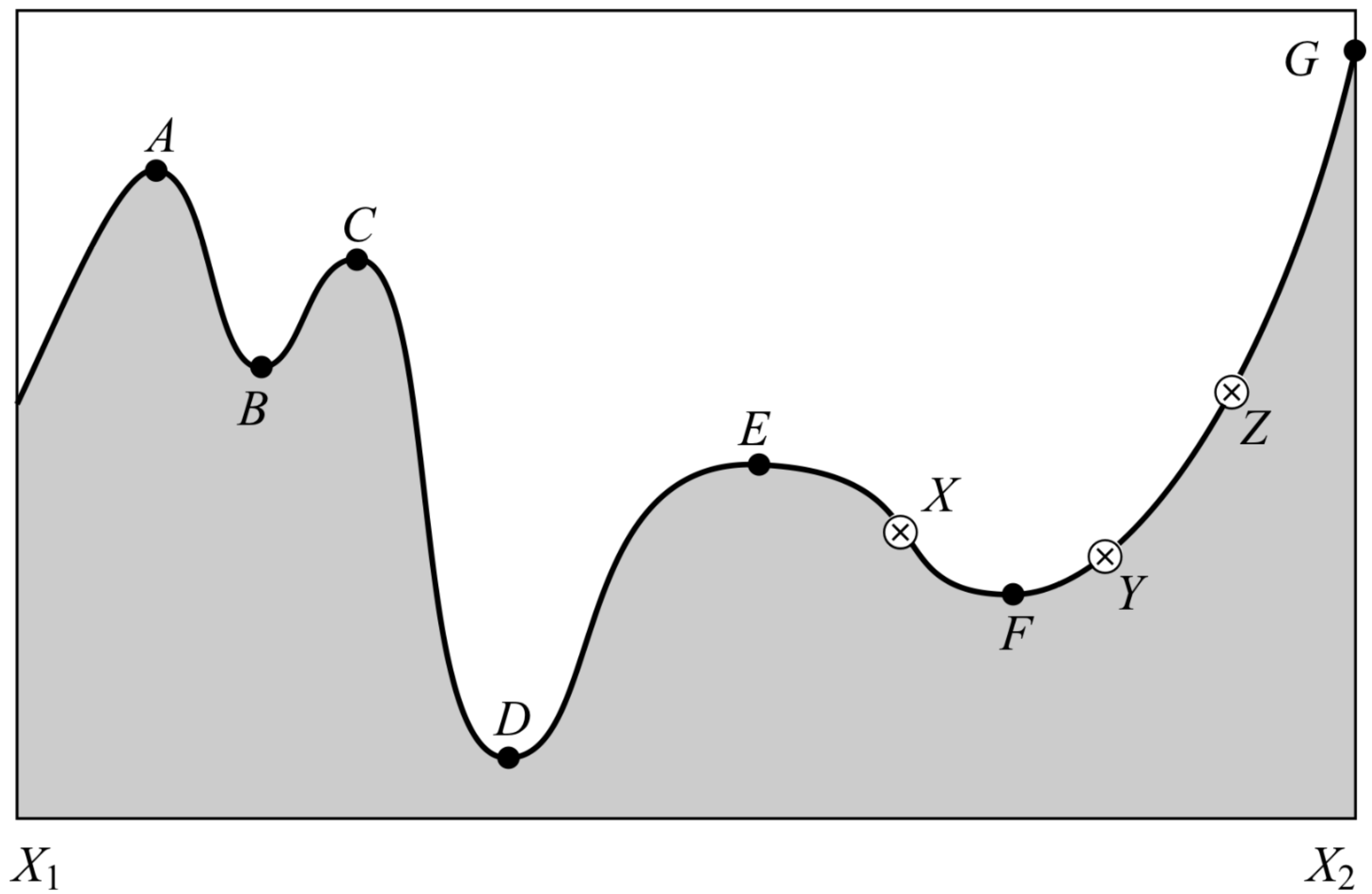


Figure 10.0.1. Extrema of a function in an interval. Points A, C, and E are local, but not global maxima. Points B and F are local, but not global minima. The global maximum occurs at G, which is on the boundary of the interval so that the derivative of the function need not vanish there. The global minimum is at D. At point E, derivatives higher than the first vanish, a situation which can cause difficulty for some algorithms. The points X, Y, and Z are said to "bracket" the minimum F, since Y is less than both X and Z.

Bracketing

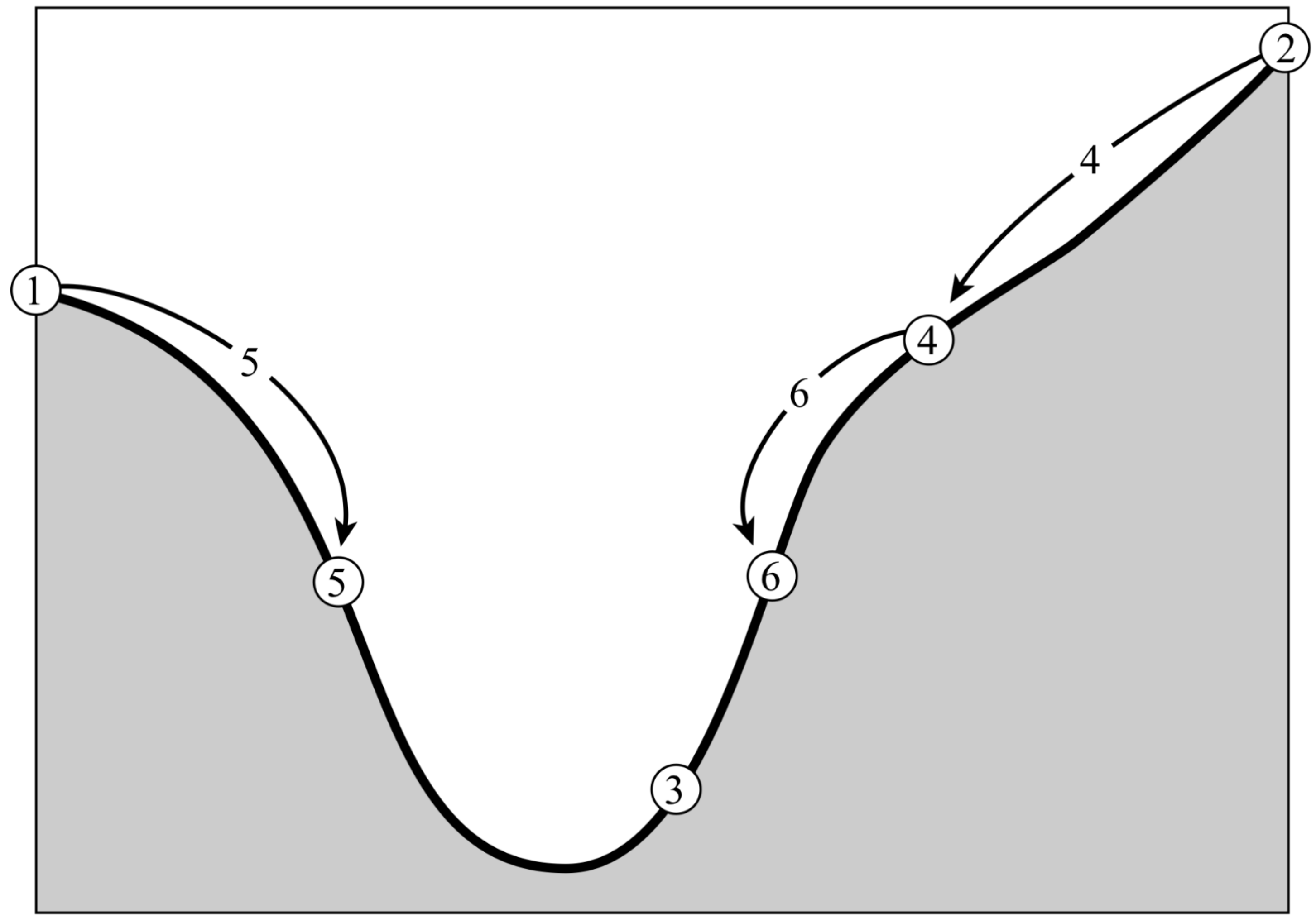


Figure 10.1.1. Successive bracketing of a minimum. The minimum is originally bracketed by points 1,3,2. The function is evaluated at 4, which replaces 2; then at 5, which replaces 1; then at 6, which replaces 4. The rule at each stage is to keep a center point that is lower than the two outside points. After the steps shown, the minimum is bracketed by points 5,3,6.

Brent's method

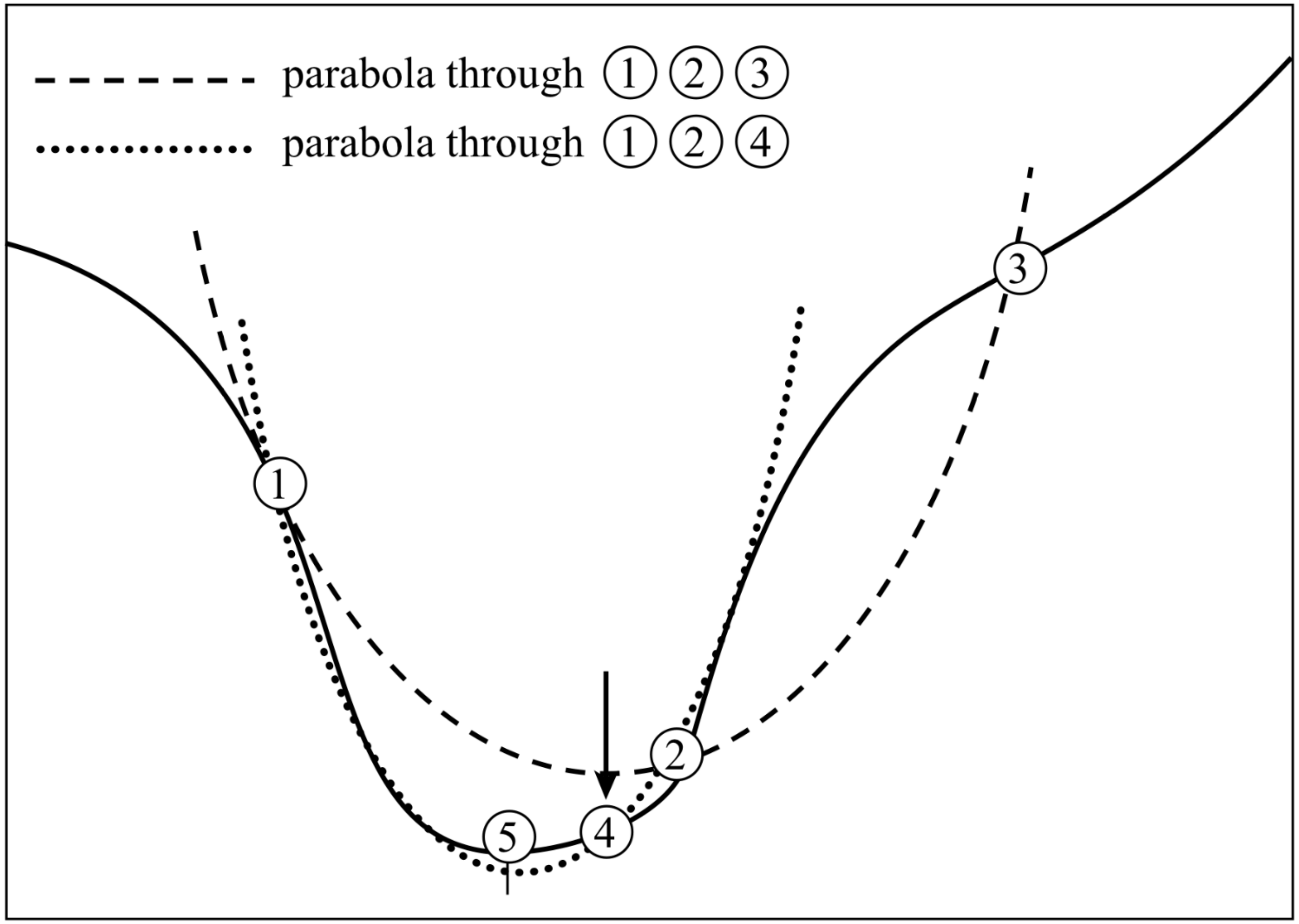


Figure 10.2.1. Convergence to a minimum by inverse parabolic interpolation. A parabola (dashed line) is drawn through the three original points 1,2,3 on the given function (solid line). The function is evaluated at the parabola's minimum, 4, which replaces point 3. A new parabola (dotted line) is drawn through points 1,4,2. The minimum of this parabola is at 5, which is close to the minimum of the function.

Brent's method

Scientific computation 4

Hyperlink list

- Toolbox [1]

http://www.cs.nyu.edu/faculty/overton/software/nlcg/index.html

- Paper [2]

http://link.springer.com/article/10.1023/A:1007618119488?LI=true

- Paper [3]

http://link.springer.com/article/10.1007/BF00940566?LI=true

Scientific computation [4]

http://linneus20.ethz.ch:8080/1_5_2.html