Root Finding

goal: sqrt of $a > 0$

$y_0 = 1$ | $A = a$

$x_0 = a$

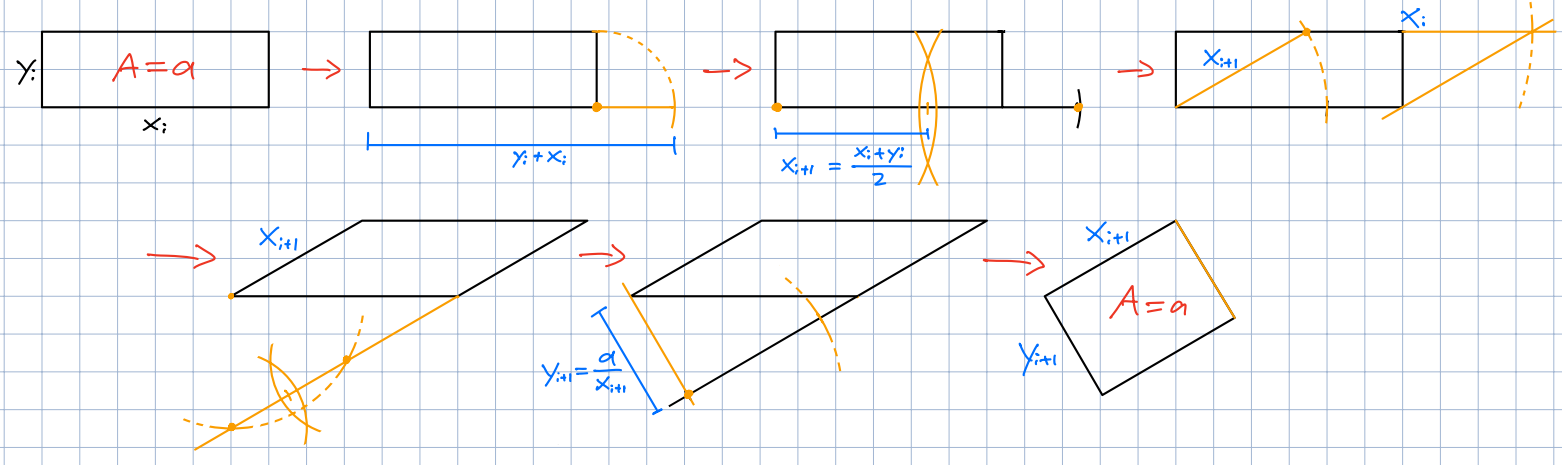
idea: successively transform conserving area until perfect square

straightedge and compass approach:

$$x_{i+1} = \frac{x_i + y_i}{2}$$

since $A = x_i \cdot y_i \overset{!}{=} a \Rightarrow y_i = \frac{a}{x_i}$ (can always express like this):

$$x_{i+1} = \frac{x_i + \frac{a}{x_i}}{2} = \frac{x_i^2 + a}{2x_i} = x_i - \frac{2x_i^2}{2x_i} + \frac{x_i^2 + a}{2x_i} = x_i - \frac{\overbrace{x_i^2 - a}^{f(x_i)}}{\underbrace{2x_i}_{f'(x_i)}}$$



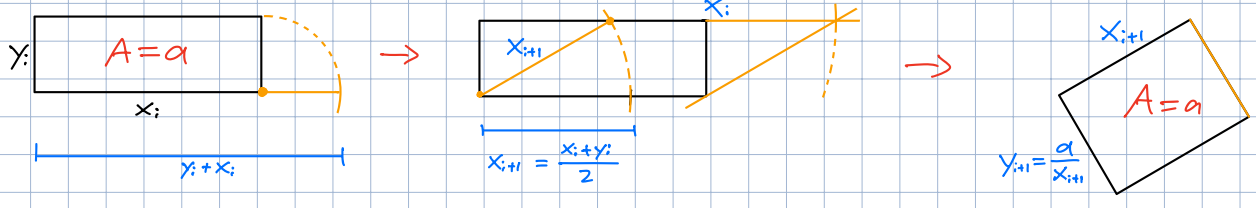
special case of Newton's Method: finding $x \in \mathbb{R}$ s.t. $f(x) = 0$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

$e_i = |x - x_i|$ decreases quadratically in i (show using Taylor Expansion)

BUT careful: can also fail badly if

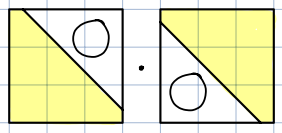
- x_0 is too far from correct solution x
- $f'(x_i) = 0$ for some x_i
- f' unbounded close to the root x



Linear Systems

LU Decomposition

using Gaussian Elimination (-> and "partial pivoting") any $\underline{A}$ can be decomposed into $\underline{A} = \underline{L}\underline{U}$, where $\underline{L}$ is a lower triangular and $\underline{U}$ is an upper triangular matrix



a linear system $\underline{A}\vec{x} = \vec{b}$ can then be solved:

1. solve $\underline{L}\vec{c} = \vec{b}$, 2. solve $\underline{U}\vec{x} = \vec{c}$ "backsubstitution"

decomposition: $O(n^3)$, backsubstitution: $O(n^2)$

sparse system: A has only $O(n)$ nonzero entries

for sparse systems often better to use an iterative method

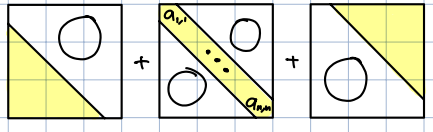
Iterative Methods

Jacobi Method

$$x_i^{(k+1)} = \frac{1}{a_{i,i}}\left(b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{i,j}\, x_j^{(k)}\right), \quad i = 1, \dots, n$$

or in matrix notation:

$$\vec{x}^{(k+1)} = \underline{D}^{-1}\left(\vec{b} - \underline{L}\vec{x}^{(k)} - \underline{U}\vec{x}^{(k)}\right), \qquad \underline{A} = \underline{L} + \underline{D} + \underline{U} =$$



can be improved if always the most recent values of x_i are used

Gauss-Seidel Method

$$x_i^{(k+1)} = \frac{1}{a_{i,i}}\left(b_i - \sum_{j=1}^{i-1} a_{i,j}\, x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{i,j}\, x_j^{(k)}\right)$$

or

$$\vec{x}^{(k+1)} = \underline{D}^{-1}\left(\vec{b} - \underline{L}\vec{x}^{(k+1)} - \underline{U}\vec{x}^{(k)}\right) \quad \Leftrightarrow \quad \vec{x}^{(k+1)} = \left(\underline{D} + \underline{L}\right)^{-1}\left(\vec{b} - \underline{U}\vec{x}^{(k)}\right)$$

advantages:
- converges faster (often)
- easier to implement, because $x_i^{(k)}$ can simply be overwritten by $x_i^{(k+1)}$

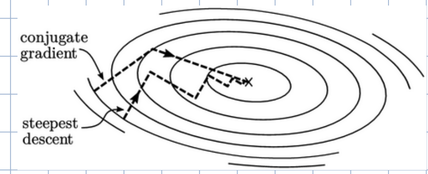
(... further improvement can be obtained by successive over-Relaxation (SOR) ...)

diagonal dominance: $|a_{i,i}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{i,j}|, \quad i = 1, \dots, n$

both methods are guaranteed to converge to correct solution if $\underline{A}$ is diagonally dominant
→ Gauss-Seidel also converges if $\underline{A}$ is spd

cost for each iteration is $O(n^2)$ and only $O(n)$ if $\underline{A}$ is sparse

Symmetric Positive-Definite Matrices



symmetric: $\underline{A}^T = \underline{A}$

positive def.: $\vec{x}^T \underline{A} \vec{x} > 0$ for $\vec{x} \neq \vec{0}$

Gauss-Seidel also converges if $\underline{A}$ is spd, but in this case an even better iterative method exists

for such matrices, the solution of $\underline{A}\vec{x} = \vec{b}$ is also the minimizer of $f(\vec{x}) = \frac{1}{2}\vec{x}^T\underline{A}\vec{x} - \vec{x}^T\vec{b}$ because $\nabla f(\vec{x}) = \frac{1}{2}(\underline{A} + \underline{A}^T) - \vec{b} \overset{\underline{A}=\underline{A}^T}{=} \underline{A}\vec{x} - \vec{b}$

Gradient Descent → simplest

$$\vec{x}_{k+1} = \vec{x}_k + \alpha_k \vec{r}_k \quad \text{with} \quad \vec{r}_k = -\nabla f(\vec{x}_k) = \vec{b} - \underline{A}\vec{x}_k$$

α_k is chosen such that $f(\vec{x}_{k+1})$ is minimal wrt α_k:

$$\alpha_k = \frac{\vec{r}_k^T \vec{r}_k}{\vec{r}_k^T \underline{A} \vec{r}_k}$$

$$f(x_{k+1}) = \frac{1}{2}\vec{x}_{k+1}^T \underline{A} \vec{x}_{k+1} - \vec{x}_{k+1}^T \vec{b}$$
$$= \frac{1}{2}(\vec{x}_k + \alpha_k \vec{r}_k)^T \underline{A} (\vec{x}_k + \alpha_k \vec{r}_k) - (\vec{x}_k + \alpha_k \vec{r}_k)^T \vec{b}$$
$$\frac{\partial f}{\partial \alpha_k} = (\underline{A}(\vec{x}_k + \alpha_k \vec{r}_k))^T \vec{r}_k - \vec{r}_k^T \vec{b} = (\underline{A}\vec{x}_k + \alpha_k \underline{A}\vec{r}_k)^T \vec{r}_k - \vec{r}_k^T\vec{b} = \vec{x}_k^T \underline{A} \vec{r}_k + \alpha_k \vec{r}_k^T \underline{A} \vec{r}_k - \vec{r}_k^T \vec{b}$$
$$\overset{!}{=} 0 \implies \vec{r}_k^T \vec{b} = \vec{x}_k^T \underline{A} \vec{r}_k + \alpha_k \vec{r}_k^T \underline{A} \vec{r}_k$$
$$\rightarrow \alpha_k = \frac{\vec{r}_k^T \vec{b} - \vec{x}_k^T \underline{A} \vec{r}_k}{\vec{r}_k^T \underline{A} \vec{r}_k} = \frac{(\vec{b}^T - \vec{x}_k^T \underline{A})\vec{r}_k}{\vec{r}_k \underline{A} \vec{r}_k} = \frac{\vec{r}_k^T \vec{r}_k}{\vec{r}_k^T \underline{A} \vec{r}_k}$$

the repeated gradients are not in the best direction
→ not orthogonal

Conjugate Gradient Descent

idea: take directions d_k which are pairwise conjugate wrt $\underline{A}$

two vectors are $\underline{A}$-conjugate if $\vec{v}^T \underline{A} \vec{u} = 0$ (note: for $\underline{A} = \underline{I}$, this is standard orthogonality)

$\vec{x}_0$, $\vec{r}_0 = \vec{b} - \underline{A}\vec{x}_0$, $\vec{d}_0 = \vec{r}_0$

then iterate

$$\vec{x}_{k+1} = \vec{x}_k + \alpha_k \vec{d}_k$$
$$\alpha_k = \vec{r}_k^T \vec{d}_k / \vec{d}_k^T \underline{A} \vec{d}_k = \vec{r}_k^T \vec{r}_k / \vec{d}_k^T \underline{A} \vec{d}_k$$
$$\vec{r}_{k+1} = \vec{b} - \underline{A}\vec{x}_{k+1} = \vec{b} - \underline{A}(\vec{x}_k + \alpha_k \vec{d}_k) = \vec{r}_k - \alpha_k \underline{A} \vec{d}_k$$
$$\vec{d}_{k+1} = \vec{r}_{k+1} + \beta_k \vec{d}_k$$
$$\beta_k = -\vec{d}_k^T \underline{A} \vec{r}_{k+1} / \vec{d}_k^T \underline{A} \vec{d}_k$$

$$f(x_{k+1}) = \frac{1}{2}\vec{x}_{k+1}^T \underline{A} \vec{x}_{k+1} - \vec{x}_{k+1}^T \vec{b}$$
$$= \frac{1}{2}(\vec{x}_k + \alpha_k \vec{d}_k)^T \underline{A} (\vec{x}_k + \alpha_k \vec{d}_k) - (\vec{x}_k + \alpha_k \vec{d}_k)^T \vec{b}$$
$$\frac{\partial f}{\partial \alpha_k} = (\underline{A}(\vec{x}_k + \alpha_k \vec{d}_k))^T \vec{d}_k - \vec{d}_k^T \vec{b} = (\underline{A}\vec{x}_k + \alpha_k \underline{A}\vec{d}_k)^T \vec{d}_k - \vec{d}_k^T\vec{b} = \vec{x}_k^T \underline{A} \vec{d}_k + \alpha_k \vec{d}_k^T \underline{A} \vec{d}_k - \vec{d}_k^T \vec{b}$$
$$\overset{!}{=} 0 \implies \vec{d}_k^T \vec{b} = \vec{x}_k^T \underline{A} \vec{d}_k + \alpha_k \vec{d}_k^T \underline{A} \vec{d}_k$$
$$\rightarrow \alpha_k = \frac{\vec{d}_k^T \vec{b} - \vec{x}_k^T \underline{A} \vec{d}_k}{\vec{d}_k^T \underline{A} \vec{d}_k} = \frac{(\vec{b}^T - \vec{x}_k^T \underline{A})\vec{d}_k}{\vec{d}_k^T \underline{A} \vec{d}_k} = \frac{\vec{r}_k^T \vec{d}_k}{\vec{d}_k^T \underline{A} \vec{d}_k}$$

Arnoldi: $AQ = QH$
Gram Schmidt
Krylov
upper Hessenberg $H = Q^{-1}AQ$

high eigenvalues not so reliable
but Ritz values

similar matrices → same eigenvalues

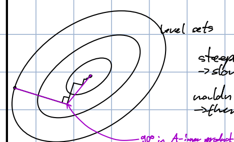
$x_k \in K_k$ (k^{th} Krylov space)
$b \in K_1$
$r_k = b - Ax_k \in K_{k+1}$

residuals are orthogonal

① $r_i^T r_k = 0 \quad i < k$ — orthogonal in usual sense (because have A already built into them)
② $(x_i - x_{i-1})^T A (x_k - x_{k-1}) = 0 \quad i < k$ — orthogonal in A-inner-product

going into cycle with new d: search direction, where Δx is going to go
α is how long along we go in this direction d
change in r is easy: if I know the change in x, I know the change in r
but now we need to look ahead: what direction we travel next? find number β and search direction
→ search direction is in r but with one correction

1. $\alpha_k = \frac{r_{k-1}^T r_{k-1}}{d_{k-1}^T A d_{k-1}}$
2. $x_k = x_{k-1} + \alpha_k d_{k-1}$
3. $r_k = r_{k-1} - \alpha_k A d_{k-1}$
4. $\beta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$
5. $d_k = r_k + \beta_k d_{k-1}$ — new direction that is A-orthogonal to ALL previous directions (②)



steepest descent frustrating
→ slow rate of convergence

wouldn't happen if contours were circles ($A = I$)
→ then only one step and we are done

$\alpha_k = \vec{d}_k^T \vec{r}_k / \vec{d}_k^T \underline{A} \vec{d}_k$ (chosen again such that $f(\vec{x}_{k+1})$ is minimal)

β_k is chosen such that $\vec{d}_{k+1}$ is $\underline{A}$-conjugate to $\vec{d}_k$: $\vec{d}_{k+1}^T \underline{A} \vec{d}_k = (\vec{r}_{k+1} + \beta_k \vec{d}_k)^T \underline{A} \vec{d}_k = \vec{r}_{k+1}^T \underline{A} \vec{d}_k + \beta_k \vec{d}_k^T \underline{A} \vec{d}_k \overset{!}{=} 0$

the following subspaces of $\mathbb{R}^n$ are equal: $\langle \vec{x}_1, \dots, \vec{x}_k \rangle = \langle \vec{r}_0, \dots, \vec{r}_{k-1} \rangle = \langle \vec{d}_0, \dots, \vec{d}_{k-1} \rangle$

residuals $\vec{r}_k$ are pairwise orthogonal: $\vec{r}_k^T \vec{r}_j = 0$ for $j < k$

directions $\vec{d}_k$ are pairwise conjugate: $\vec{d}_k^T \underline{A} \vec{d}_j = 0$ for $j < k$

residual $\vec{r}_{k+1}$ is orthogonal to $\vec{d}_k$ since $\vec{r}_{k+1}^T \vec{d}_k = -(\vec{x}_{k+1}^T \underline{A}^T - \vec{b}^T)\vec{d}_k = -\frac{\partial f}{\partial \alpha_k} \overset{\text{by construction}}{=} 0$

$\vec{r}_k$ also orthogonal to previous $\vec{d}_i$'s: $\vec{r}_k^T \vec{d}_j = 0$ for $j < k$
→ makes sense, since?

$\vec{d}_k$ is a linear combination of the residuals $\{\vec{r}_0, \dots, \vec{r}_k\}$: $\vec{d}_k = \sum_{i=0}^{k} \left(\prod_{j=i+1}^{k} \beta_j \right) \vec{r}_i$

Polynomial Interpolation

most common way of expressing polynomial is to use monomial basis: $\{1, x, x^2, \dots, x^n\}$

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

can be evaluated efficiently using Horner's scheme in $O(n)$ operations:

```
p := a_n
for i from n-1 to 0 do
    p := p x + a_i
return p
```

Polynomial Curves $\vec{P}: \mathbb{R} \to \mathbb{R}^d$

$$\vec{P}(t) = \vec{A}_n t^n + \vec{A}_{n-1} t^{n-1} + \cdots + \vec{A}_1 t + \vec{A}_0$$

Horner also works etc...

often interested in interpolating polynomial curve $\vec{P}(t_i) = \vec{P}_i, \quad i = 0, 1, \dots, n$

can be written as linear system

$$\underline{V}\,\underline{A} = \underline{P} \; ; \quad \underline{V} = \begin{bmatrix} 1 & t_0 & \cdots & t_0^n \\ 1 & t_1 & \cdots & t_1^n \\ \vdots & \vdots & & \vdots \\ 1 & t_n & \cdots & t_n^n \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}, \quad \underline{A} = \begin{bmatrix} — & \vec{A}_0^T & — \\ — & \vec{A}_1^T & — \\ & \vdots & \\ — & \vec{A}_n^T & — \end{bmatrix} \in \mathbb{R}^{(n+1)\times d} \quad \underline{P} = \begin{bmatrix} — & \vec{P}_0^T & — \\ — & \vec{P}_1^T & — \\ & \vdots & \\ — & \vec{P}_n^T & — \end{bmatrix} \in \mathbb{R}^{(n+1)\times d}$$

Vandermonde