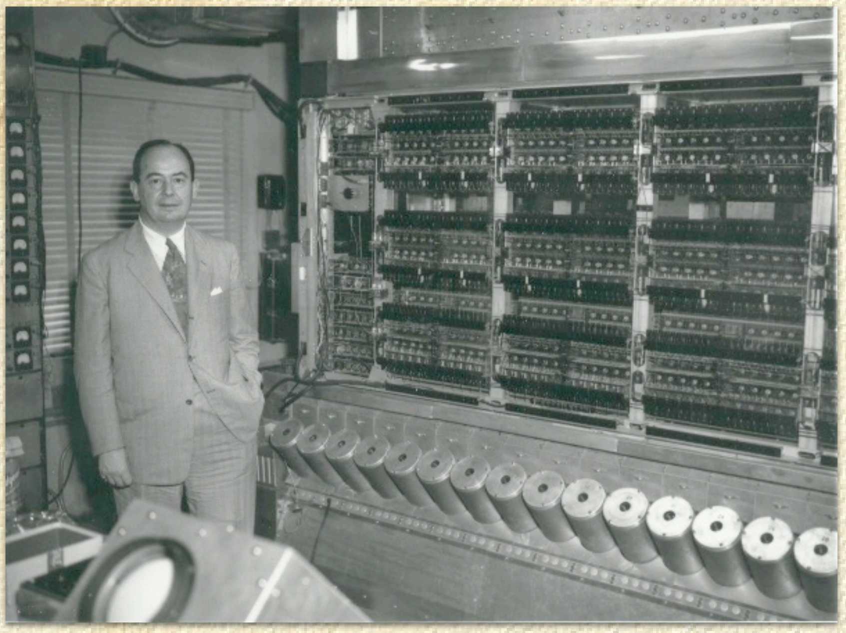
Geometric optimization of the evaluation of finite element matrices

Robert C. Kirby
Texas Tech University

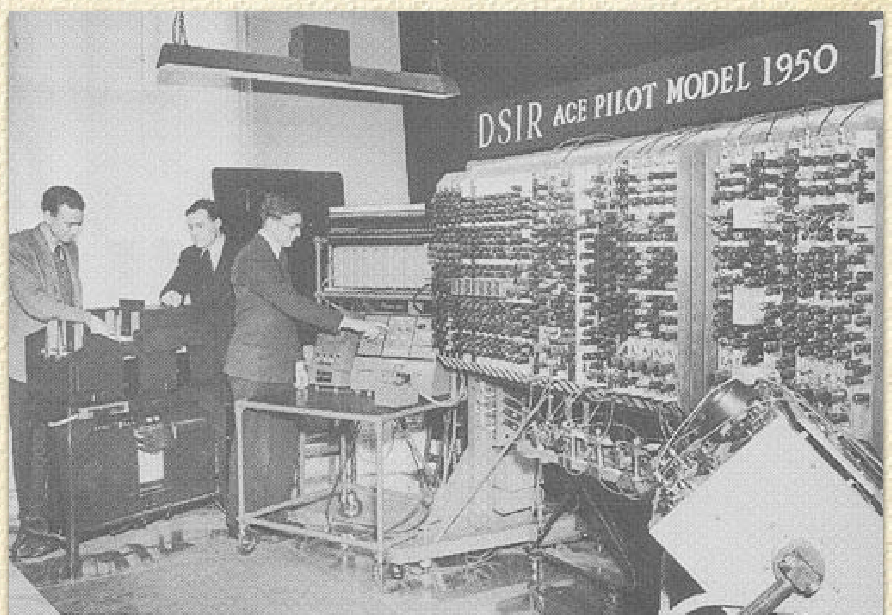
A new day dawns...



Numerical analysis

Theoretical CS

Computer engineering



KFLOPS

Cryptography

Turbulence

Software makes it happen

- Canonical problems
- Parallel libraries
- Performance optimization
- PETSc, Trilinos, HYPRE
- SciDAC/TOPS (DOE)

$$Ax = b$$

$$Ax = \lambda Bx$$

$$F(x) = 0$$

$$f(\dot{x}, x, t, p) = 0$$

$$\min_u \phi(x, u) s.t. \; F(x, u) = 0$$

It's a big gap...

Fusion

Biomechanics

Turbulence

etc...

http://palodurocanyon.com

Linear algebra

Newton

Multigrid

MPI

The “Grant” Method

- Immense resources
- Victory at any cost
- Patient, decisive
- Never quit/let up

I need an ARMY!

$$B_t = \nabla \times (u \times B)$$
$$u_t + u \cdot \nabla u - \mu \Delta u + \nabla p = (\nabla \times B) \times B$$
$$\nabla \cdot u = 0$$
$$\nabla \cdot B = 0$$

...and this is still a *model*!

Too often...

Victory through superior cannon fodder

Another way to see it

- Think like von Neumann, Turing...build a machine.
- ...and spin new mathematical opportunities

Example: Laplacian

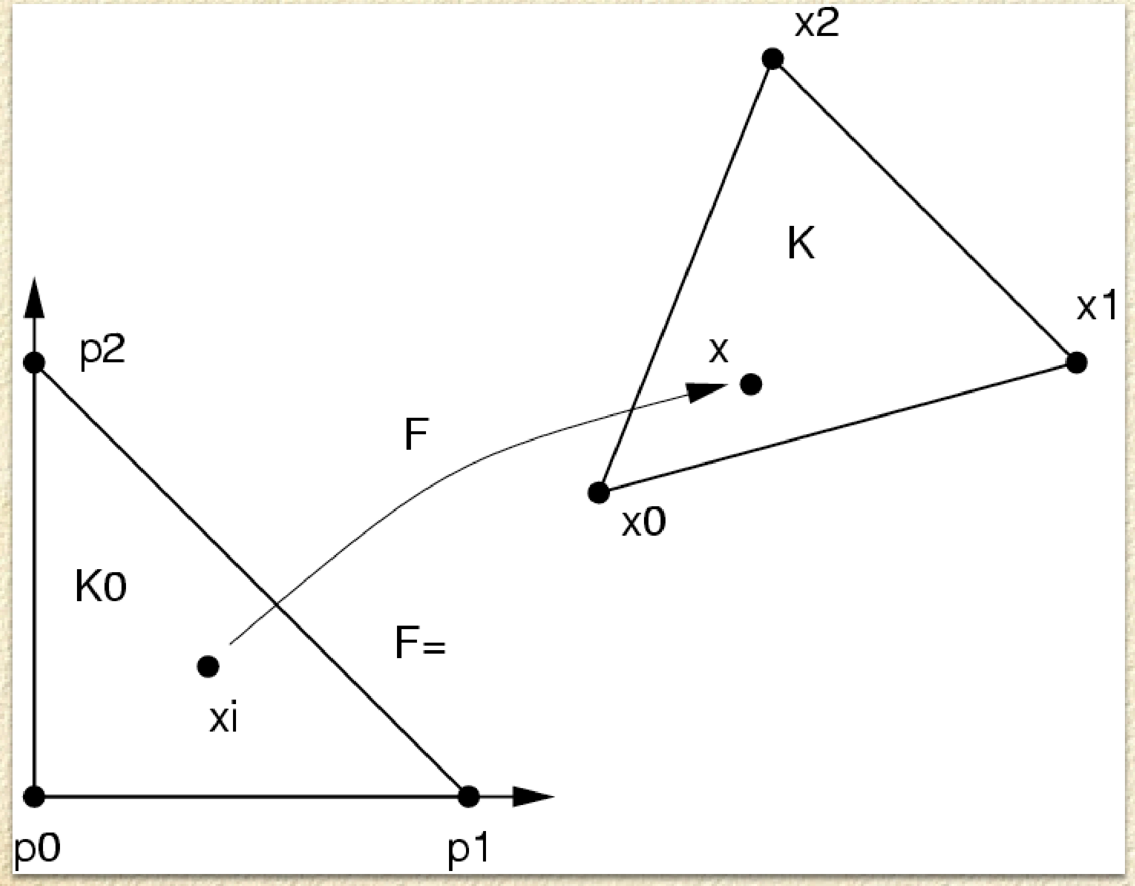
- Weak form:

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v$$

- Element matrix:

$$\begin{aligned} A_i^K &= \int_K \nabla \phi_{i_1} \cdot \nabla \phi_{i_2} \mathrm{d}x \\ &= \sum_{d=1}^D \int_K D_x^d \phi_{i_1} D_x^d \phi_{i,2} \mathrm{d}x \end{aligned}$$

Element transformation



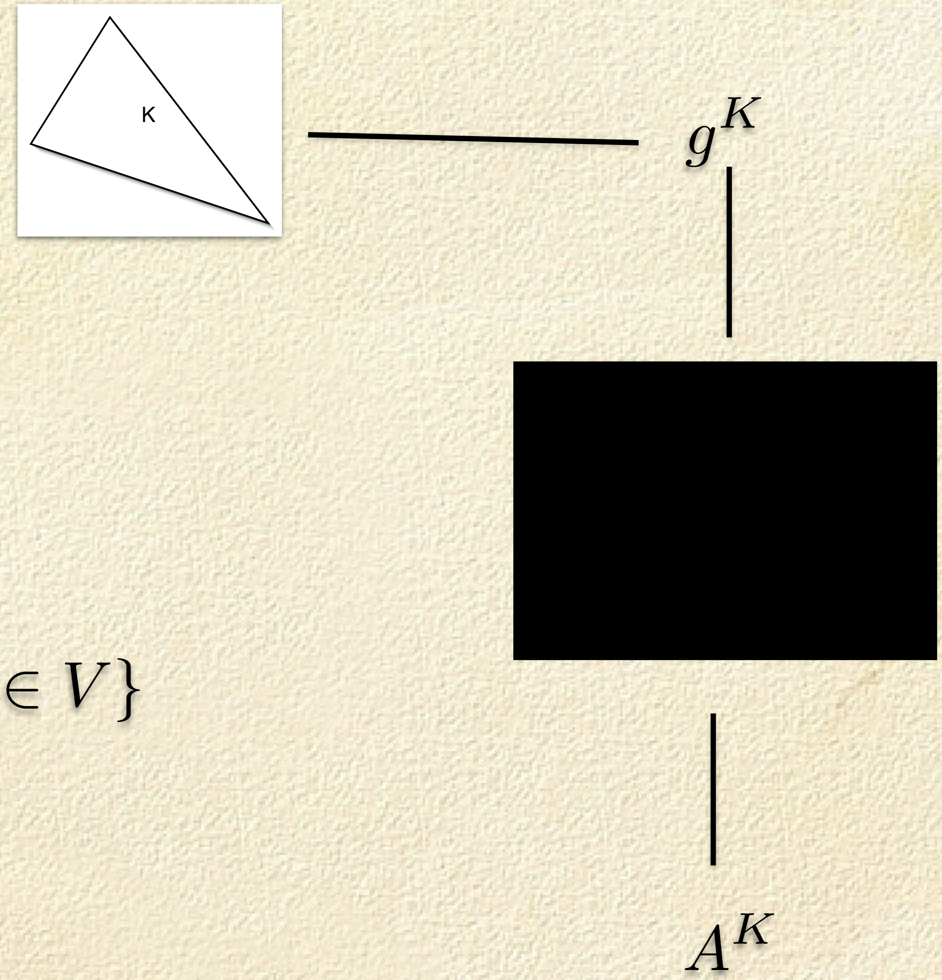
Structure of computation

$$
\begin{aligned}
A_i^K &= \int_K \nabla\phi_{i_1}^{K,1}(x)\cdot\nabla\phi_{i_2}^{K,2}(x)\mathrm{d}x \\
&= \det F_K' \sum_\beta \frac{\partial X_{\alpha_1}}{\partial x_\beta}\frac{\partial X_{\alpha_2}}{\partial x_\beta}\int_{K_0}\frac{\partial\Phi_{i_1}^1(X)}{\partial X_{\alpha_1}}\frac{\partial\Phi_{i_2}^2(X)}{\partial X_{\alpha_2}}\mathrm{d}X \\
&= \sum_\alpha A_{i\alpha}^0 G_K^\alpha
\end{aligned}
$$

Matvec

- Reference element/geometry separated
- Contraction happens on each triangle
- Local matrix inserted into global matrix

Abstract Problem



- $V \subset \mathbb{R}^d$ fixed
- $g \in \mathbb{R}^d$ arbitrary
- Compute $\{v^t g : v \in V\}$

A for Poisson (k=2,d=2)

3	0	0	-1	1	1	-4	-4	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	3	1	1	0	0	4	0	-4	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
-4	0	0	0	-4	-4	8	4	0	-4	0	4
-4	0	0	0	0	0	4	8	-4	-8	4	0
0	0	0	4	0	0	0	-4	8	4	-8	-4
4	0	0	0	0	0	-4	-8	4	8	-4	0
0	0	0	-4	0	0	0	4	-8	-4	8	4
0	0	0	-4	-4	-4	4	0	-4	0	4	8

Canonical Form

$$A_i^K = \sum_{\gamma \in \mathcal{C}} \int_K \prod_{j=1}^{m} c_j(i,\gamma) D_x^{\delta_j(i,\gamma)} \phi_{\iota_j(i,\gamma)}^{K,j}[\kappa_j(i,\gamma)] \mathrm{d}x$$

$c_j(i,\gamma)$	coefficient
$\iota_j(i,\gamma)$	basis function
$\kappa_j(i,\gamma)$	vector component
$\delta_j(i,\gamma)$	derivative multiindex

Example

$$a(v,u) = \sum_{\gamma_1=1}^{d}\sum_{\gamma_2=1}^{d}\int_\Omega w \frac{\partial v[\gamma_1]}{\partial x_{\gamma_2}}\frac{\partial u[\gamma_1]}{\partial x_{\gamma_2}}\mathrm{d}x$$

$$A_i^K = \sum_{\gamma_1=1}^{d}\sum_{\gamma_2=1}^{d}\sum_{\gamma_3=1}^{|V_3^K|}\int_\Omega \frac{\partial \phi_{i_1}^{K,1}[\gamma_1]}{\partial x_{\gamma_2}}\frac{\partial \phi_{i_2}^{K,2}[\gamma_1]}{\partial x_{\gamma_2}} w_{\gamma_3}\phi_{\gamma_3}^{K,3}\mathrm{d}x,$$

r	2
m	3
$\iota(i,\gamma)$	(i_1, i_2, γ_3)
$\delta(i,\gamma)$	$(\gamma_2, \gamma_2, \emptyset)$
$\kappa(i,\gamma)$	$(\gamma_1, \gamma_2, \emptyset)$
$c_j(i,\gamma)$	$(1, 1, w_{\gamma_3})$

Representation Theorem

$$A_i^K = \sum_{\alpha \in \mathcal{A}} A_{i\alpha}^0 G_K^\alpha \quad \forall i \in \mathcal{I}$$

$$A_{i\alpha}^0 = \sum_{\beta \in \mathcal{B}} \int_{K_0} \prod_{j=1}^m D_X^{\delta'_j(i,\alpha,\beta)} \Phi_{\iota_j(i,\alpha,\beta)}^{K_0,j}[\kappa_j(i,\alpha,\beta)] \mathrm{d}X$$

$$G_K^\alpha = \sum_{\beta \in \mathcal{B}'} \det F'_K \prod_{j=1}^m c_j(i,\alpha,\beta) \prod_{j'=1}^m \prod_{k=1}^{|\delta_{j'}(i,\alpha,\beta)|} \frac{\partial X_{\delta'_{j'k}(i,\alpha,\beta)}}{\partial x_{\delta_{j'k}(i,\alpha,\beta)}}$$

Basis for FFC[*K*L]

Optimizing evaluation

3	0	0	-1	1	1	-4	-4	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	3	1	1	0	0	4	0	-4	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
-4	0	0	0	-4	-4	8	4	0	-4	0	4
-4	0	0	0	0	0	4	8	-4	-8	4	0
0	0	0	4	0	0	0	-4	8	4	-8	-4
4	0	0	0	0	0	-4	-8	4	8	-4	0
0	0	0	-4	0	0	0	4	-8	-4	8	4
0	0	0	-4	-4	-4	4	0	-4	0	4	8

Optimizing evaluation

3	0	0	-1	1	1	-4	-4	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	3	1	1	0	0	4	0	-4	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
-4	0	0	0	-4	-4	8	4	0	-4	0	4
-4	0	0	0	0	0	4	8	-4	-8	4	0
0	0	0	4	0	0	0	-4	8	4	-8	-4
4	0	0	0	0	0	-4	-8	4	8	-4	0
0	0	0	-4	0	0	0	4	-8	-4	8	4
0	0	0	-4	-4	-4	4	0	-4	0	4	8

Zero

Optimizing evaluation

3	0	0	-1	1	1	-4	-4	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	3	1	1	0	0	4	0	-4	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
-4	0	0	0	-4	-4	8	4	0	-4	0	4
-4	0	0	0	0	0	4	8	-4	-8	4	0
0	0	0	4	0	0	0	-4	8	4	-8	-4
4	0	0	0	0	0	-4	-8	4	8	-4	0
0	0	0	-4	0	0	0	4	-8	-4	8	4
0	0	0	-4	-4	-4	4	0	-4	0	4	8

Optimizing evaluation

3	0	0	-1	1	1	-4	-4	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	3	1	1	0	0	4	0	-4	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
-4	0	0	0	-4	-4	8	4	0	-4	0	4
-4	0	0	0	0	0	4	8	-4	-8	4	0
0	0	0	4	0	0	0	-4	8	4	-8	-4
4	0	0	0	0	0	-4	-8	4	8	-4	0
0	0	0	-4	0	0	0	4	-8	-4	8	4
0	0	0	-4	-4	-4	4	0	-4	0	4	8

Sparse

Optimizing evaluation

3	0	0	-1	1	1	-4	-4	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	3	1	1	0	0	4	0	-4	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
-4	0	0	0	-4	-4	8	4	0	-4	0	4
-4	0	0	0	0	0	4	8	-4	-8	4	0
0	0	0	4	0	0	0	-4	8	4	-8	-4
4	0	0	0	0	0	-4	-8	4	8	-4	0
0	0	0	-4	0	0	0	4	-8	-4	8	4
0	0	0	-4	-4	-4	4	0	-4	0	4	8

Optimizing evaluation

3	0	0	-1	1	1	-4	-4	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	3	1	1	0	0	4	0	-4	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
-4	0	0	0	-4	-4	8	4	0	-4	0	4
-4	0	0	0	0	0	4	8	-4	-8	4	0
0	0	0	4	0	0	0	-4	8	4	-8	-4
4	0	0	0	0	0	-4	-8	4	8	-4	0
0	0	0	-4	0	0	0	4	-8	-4	8	4
0	0	0	-4	-4	-4	4	0	-4	0	4	8

Equal

Optimizing evaluation

3	0	0	-1	1	1	-4	-4	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	3	1	1	0	0	4	0	-4	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
-4	0	0	0	-4	-4	8	4	0	-4	0	4
-4	0	0	0	0	0	4	8	-4	-8	4	0
0	0	0	4	0	0	0	-4	8	4	-8	-4
4	0	0	0	0	0	-4	-8	4	8	-4	0
0	0	0	-4	0	0	0	4	-8	-4	8	4
0	0	0	-4	-4	-4	4	0	-4	0	4	8

Optimizing evaluation

3	0	0	-1	1	1	-4	-4	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	3	1	1	0	0	4	0	-4	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
-4	0	0	0	-4	-4	8	4	0	-4	0	4
-4	0	0	0	0	0	4	8	-4	-8	4	0
0	0	0	4	0	0	0	-4	8	4	-8	-4
4	0	0	0	0	0	-4	-8	4	8	-4	0
0	0	0	-4	0	0	0	4	-8	-4	8	4
0	0	0	-4	-4	-4	4	0	-4	0	4	8

Colinear

Optimizing evaluation

3	0	0	-1	1	1	-4	-4	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	3	1	1	0	0	4	0	-4	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
-4	0	0	0	-4	-4	8	4	0	-4	0	4
-4	0	0	0	0	0	4	8	-4	-8	4	0
0	0	0	4	0	0	0	-4	8	4	-8	-4
4	0	0	0	0	0	-4	-8	4	8	-4	0
0	0	0	-4	0	0	0	4	-8	-4	8	4
0	0	0	-4	-4	-4	4	0	-4	0	4	8

Optimizing evaluation

3	0	0	-1	1	1	-4	-4	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	3	1	1	0	0	4	0	-4	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
-4	0	0	0	-4	-4	8	4	0	-4	0	4
-4	0	0	0	0	0	4	8	-4	-8	4	0
0	0	0	4	0	0	0	-4	8	4	-8	-4
4	0	0	0	0	0	-4	-8	4	8	-4	0
0	0	0	-4	0	0	0	4	-8	-4	8	4
0	0	0	-4	-4	-4	4	0	-4	0	4	8

Linear Combination

Optimizing evaluation

3	0	0	-1	1	1	-4	-4	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	3	1	1	0	0	4	0	-4	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
-4	0	0	0	-4	-4	8	4	0	-4	0	4
-4	0	0	0	0	0	4	8	-4	-8	4	0
0	0	0	4	0	0	0	-4	8	4	-8	-4
4	0	0	0	0	0	-4	-8	4	8	-4	0
0	0	0	-4	0	0	0	4	-8	-4	8	4
0	0	0	-4	-4	-4	4	0	-4	0	4	8

Optimizing evaluation

3	0	0	-1	1	1	-4	-4	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	3	1	1	0	0	4	0	-4	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
-4	0	0	0	-4	-4	8	4	0	-4	0	4
-4	0	0	0	0	0	4	8	-4	-8	4	0
0	0	0	4	0	0	0	-4	8	4	-8	-4
4	0	0	0	0	0	-4	-8	4	8	-4	0
0	0	0	-4	0	0	0	4	-8	-4	8	4
0	0	0	-4	-4	-4	4	0	-4	0	4	8

Edit Distance

Optimizing evaluation

3	0	0	-1	1	1	-4	-4	0	4	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
-1	0	0	3	1	1	0	0	4	0	-4	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
1	0	0	1	3	3	-4	0	0	0	0	-4
-4	0	0	0	-4	-4	8	4	0	-4	0	4
-4	0	0	0	0	0	4	8	-4	-8	4	0
0	0	0	4	0	0	0	-4	8	4	-8	-4
4	0	0	0	0	0	-4	-8	4	8	-4	0
0	0	0	-4	0	0	0	4	-8	-4	8	4
0	0	0	-4	-4	-4	4	0	-4	0	4	8

Topology and flop count

Hamming distance	$H^{+}(u,v) = \|\{i : u_i \neq v_i\}\|$
Negated Hamming distance	$H^{-}(u,v) = \|\{i : u_i \neq -v_i\}\|$
Colinearity	$u = \alpha v \rightarrow c(u,v) = 1$

Topology and flop count

Hamming distance	$H^+(u,v) = \|\{i : u_i \neq v_i\}\|$
Negated Hamming distance	$H^-(u,v) = \|\{i : u_i \neq -v_i\}\|$
Colinearity	$u = \alpha v \rightarrow c(u,v) = 1$

$$\rho(u,v) = k$$

$$\Rightarrow$$

$$u^t g = f(v^t g)$$ (k MAPs)

Topology and flop count

Hamming distance	$H^+(u,v) = \lvert\{i : u_i \neq v_i\}\rvert$
Negated Hamming distance	$H^-(u,v) = \lvert\{i : u_i \neq -v_i\}\rvert$
Colinearity	$u = \alpha v \rightarrow c(u,v) = 1$

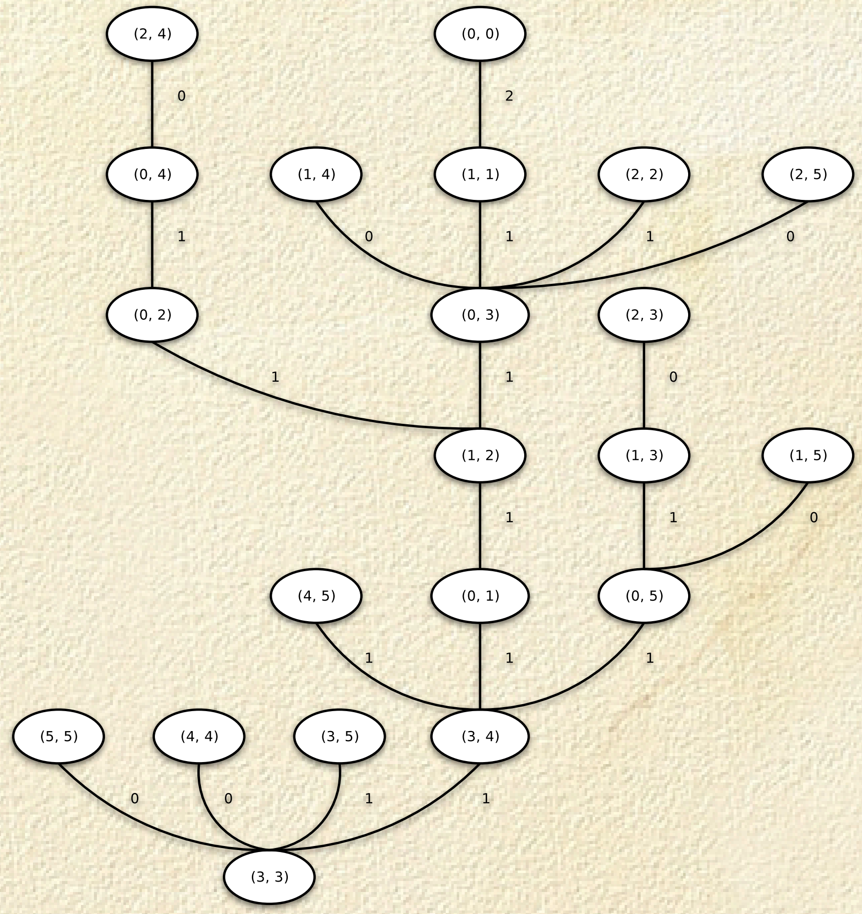
$$\rho(u,v) = k$$

"Complexity-reducing relations"

$$u^t g = f(v^t g)$$ (k MAPs)

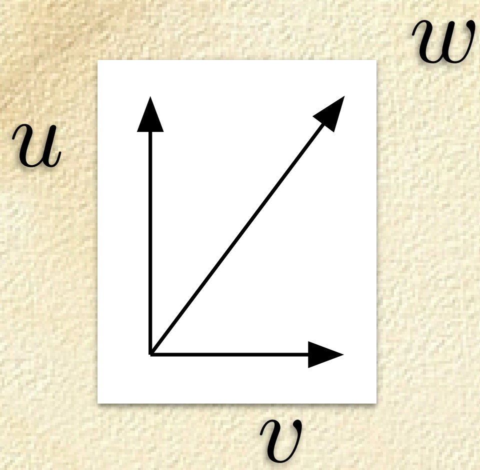
"Optimal" algorithm

- Do a dot product
- Dot product of vector "nearest" to finished vectors
- Repeat till done
- Prim's algorithm



Thm: Minimum spanning tree encodes optimal algorithm

Geometric relations



$$w = \alpha u + \beta v$$

$$w^t g = \alpha(u^t g) + \beta(v^t g)$$

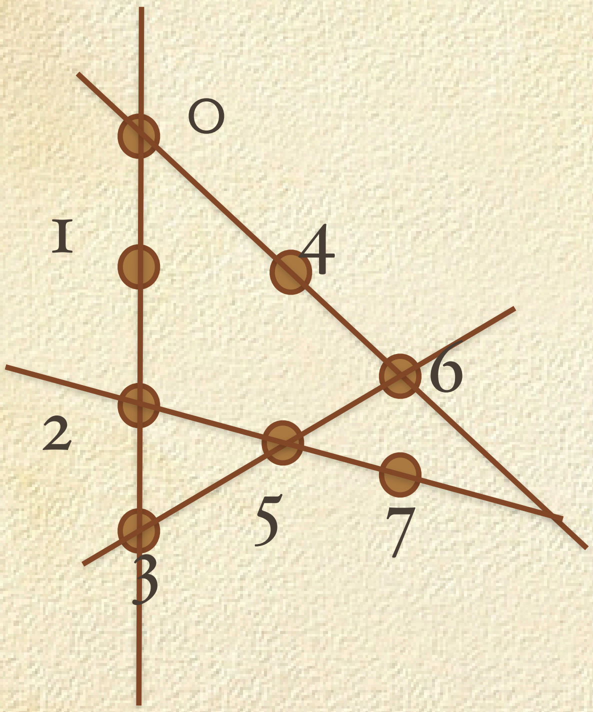
2 MAPs

- NOT a graph/ metric space!
- Low-dimensional matters most
- Detection?

Detection Algorithm

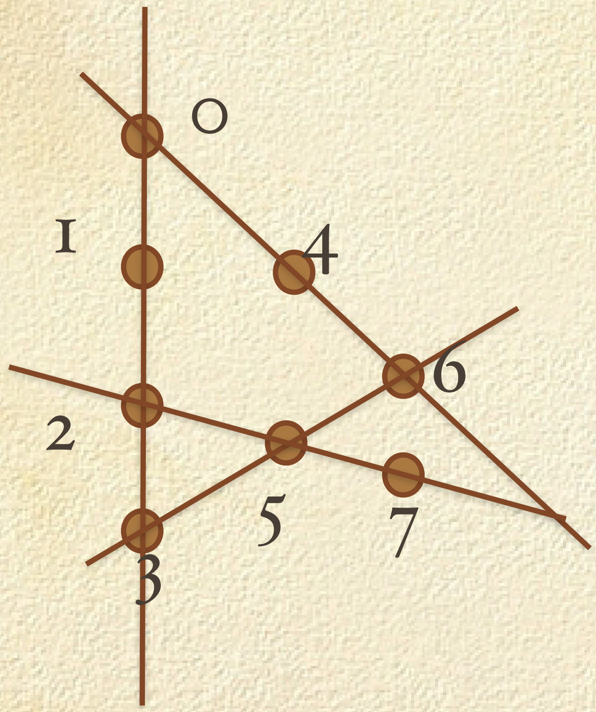
- $\mathcal{O}(d^2 N^3)$ via Gaussian elimination
- Better: Project all vectors into $\mathbb{R}^3$
  - Form all pairs of projected vectors
  - Search for colinearity among normal to planes (necessary for coplanarity)
  - Runtime: $\mathcal{O}(N^{2+\epsilon})$

Finite Linear Space



- Based on incidence between "points" and "lines"
- "Points": vectors
- "Lines": planes
- Cf. finite geometry

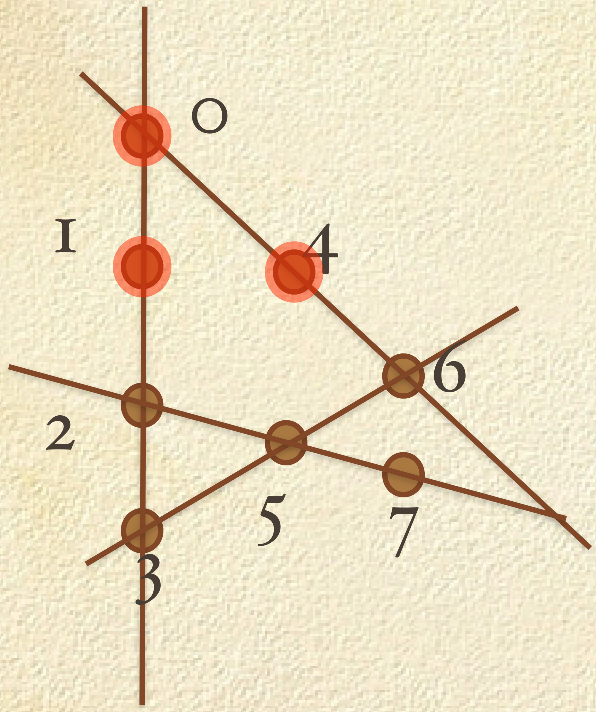
Closure



$$v \in S \rightarrow v \in \bar{S}$$

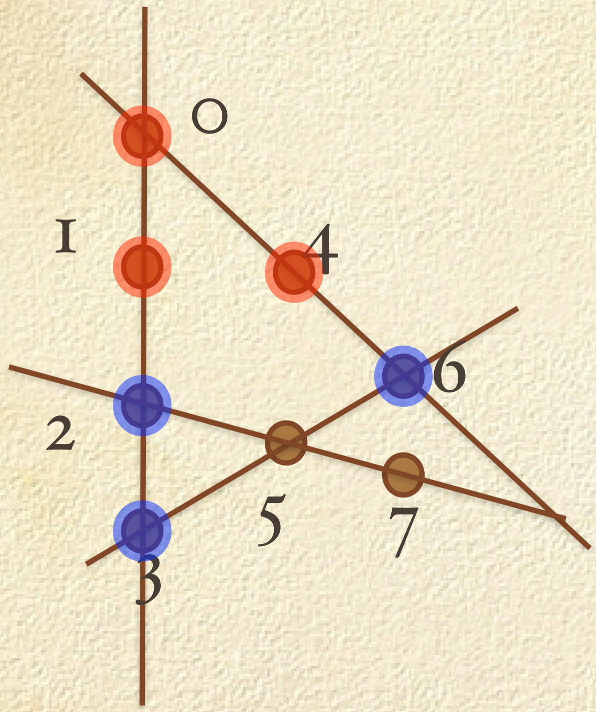
$$R(a, b, c) : a, b \in \bar{S} \rightarrow c \in \bar{S}$$

Generator



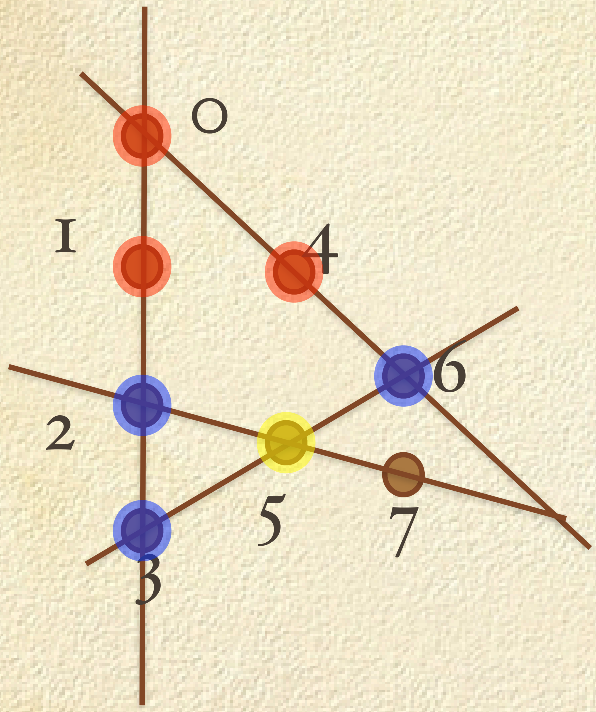
- e.g. {0,1,4}
- Only do explicit dot products for generator
- cf. Dijkstra/Prim on “line graph”

Generator



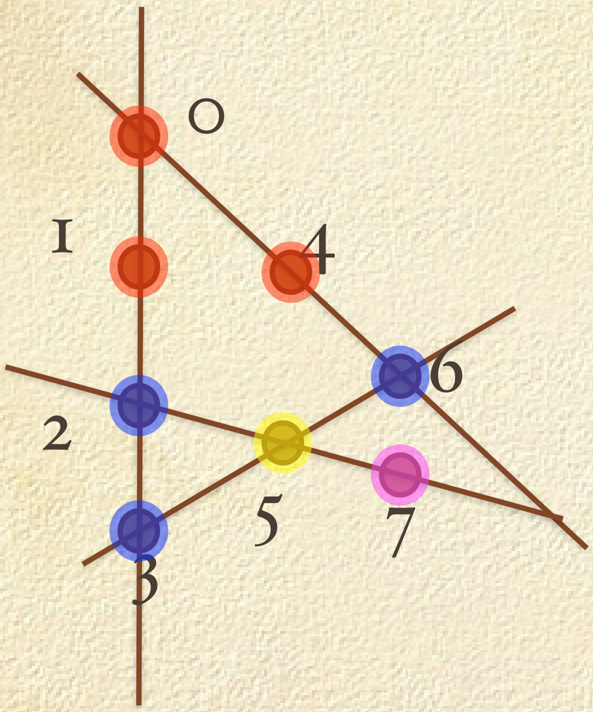
- e.g. {0,1,4}
- Only do explicit dot products for generator
- cf. Dijkstra/Prim on "line graph"

Generator



- e.g. {0,1,4}
- Only do explicit dot products for generator
- cf. Dijkstra/Prim on “line graph”

Generator



- e.g. {0,1,4}
- Only do explicit dot products for generator
- cf. Dijkstra/Prim on "line graph"

FLOP reduction

k	n	nz	ne	nc	nr	MAPs
2	55	0	6	0	6	97
3	210	0	42	22	7	329
4	630	0	150	48	67	1117

Beats CRR!

- Effective
- Increased compile-time expense
- Combined?

Combinatorial optimization problem

$\{v_i\}_{i=1}^{|V|}$ ordering of vectors

$$w_R(v_i) = \begin{cases} 2, & \exists j, k < i : R(\{v_i, v_j, v_k\}) \\ m, & otherwise \end{cases}$$

$$w_\rho(v_i) = \min_{j<i} \rho(v_i, v_j)$$

$$w(v_i) = \min(w_{R,i}, w_{\rho,i})$$

$$\min_{\mathcal{V}} \sum_{i=1}^{|V|} w(v_i)$$

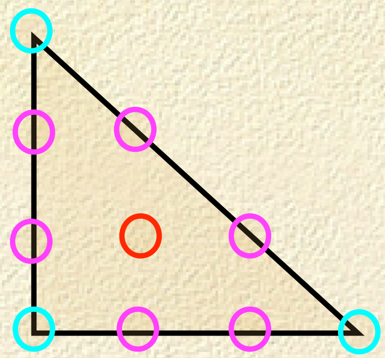
On the horizon for form evaluation

$$A = \prod A_i$$

sparse?

$$u \mapsto \nabla u|_\xi$$

fast evaluation?



barycentric groups?

- Algebraic theory? (Püschel for signal processing)
- Automate spectral elements?