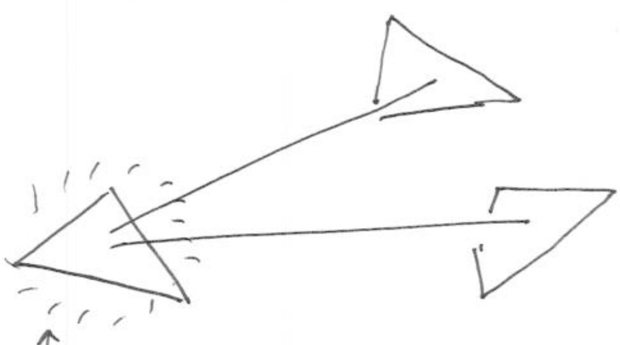
Iterative Methods in Numerical L.A.

We can be faced with linear algebra problems which are very large and very sparse - certainly too big to handle by standard methods.

Example. The Radiosity Problem in computer graphics. How is "ambient" light distributed in a scene?



~~patch~~
triangle i

Let B_i = light energy leaving triangle i.
Then

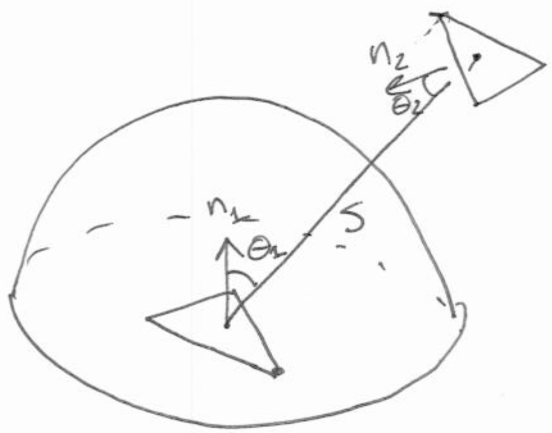
$$B_i = E_i + R_i \sum_{j=1}^{n} B_j F_{ij}$$

where

E_i = energy emitted by triangle i

R_i = reflectivity of ~~patch~~ triangle i

F_{ij} = a "view factor" describing the solid angle subtended by triangle j in triangle i's sphere of view directions.



$$F_{12} = \frac{\cos\theta_1 \cos\theta_2}{\pi S^2} \cdot A_1 A_2$$

where A_1, A_2 are the areas of the triangles. Note that $F_{ij} = 0$ if something is in the way!

In a rendering problem, we Know E_i (some of which better be nonzero or there will be no light in the scene!) and the F_{ij} and R_i and are trying to solve for the B_i.

We write this as

$$E_i = B_i - R_i \sum_{j=1}^{n} B_j F_{ij}$$

or

$$\vec{E} = \left(I - \begin{bmatrix} R_1 & & 0 \\ & \ddots & \\ 0 & & R_n \end{bmatrix} \begin{bmatrix} F_{ij} \end{bmatrix} \right) \vec{B}$$

In a real scene* (for instance, Toy Story 3) is rendered with radiosity throughout, there are a few million triangles, the F_{ij} are very small, and the R_i are (of course) $\ll 1$.

We can sum up this type of problem:

Solve $Ax = b$.

A is huge, sparse and <u>diagonally dominant</u>.

as a good candidate for an <u>iterative</u> method.

Definition. Given A, b, x_0 an iterative method generates a sequence $x_i \to A^{-1}b$.

Definition. A splitting of A is a decomposition $A = M - K$ with M nonsingular.

Given a splitting, we write

$$Ax = Mx - Kx = b$$

$$\Rightarrow \quad Mx = Kx + b$$

$$\Rightarrow \quad x = M^{-1}Kx + M^{-1}b$$

We then have

$$R := M^{-1}K, \quad c := M^{-1}b$$

and define our iterative method by

$$x_{m+1} = R x_m + c$$

If $x_{m+1} = x_m$, then we can check that they solve $Ax = b$ as expected. But when does the sequence converge?

Lemma. Let $\| \ \|$ be an operator norm for matrices. If $\|R\| < 1$ then $x_{m+1} = R x_m + c$ converges in the corresponding vector norm.

Proof. We know if x solves $Ax = b$ then $x = Rx + c$. So compute

$$x_{m+1} - x = (R x_m + c) - (Rx + c) = R(x_m - x).$$

So

$$\|x_{m+1} - x\| \leq \|R\| \, \|x_m - x\| \leq \|R\|^{m+1} \|x_0 - x\| \; \square.$$

This convergence criterion doesn't seem very natural (~~why~~ if our preferred operator norm doesn't work, how do we know if another will?).
So we now replace it with one that makes more sense.

Definition. The spectral radius of R is $\rho(R) = \max |\lambda|$ where λ is an eigenvalue of R.

Lemma. For any operator norm, $\rho(R) \le \|R\|$.
Given an R, $\epsilon > 0$ there is an operator norm (depending on R, ϵ) so that

$$\|R\|_* \le \rho(R) + \epsilon.$$

Proof. Suppose x_λ is an eigenvector corresponding to λ. Then

$$\|R\| = \max_x \frac{\|Rx\|}{\|x\|} \ge \frac{\|Rx_\lambda\|}{\|x_\lambda\|} \ge |\lambda|.$$

To build a norm, take S so that

$$S^{-1} R S = J$$

is the Jordan form of R, or

$$J = \begin{bmatrix} \lambda_1 & 1 & & & & \\ & \ddots & 1 & & & \\ & & \lambda_1 & 0 & & \\ & & & \lambda_2 & & \\ & & & & \ddots & \\ & & & & & \lambda_n \end{bmatrix}$$

we then let

$$D_\epsilon = \begin{bmatrix} 1 & & & \\ & \epsilon & & \\ & & \epsilon^2 & \\ & & & \ddots \\ & & & & \epsilon^n \end{bmatrix}$$

Now

$$D_\epsilon^{-1} J D_\epsilon = D_\epsilon^{-1} \begin{bmatrix} \lambda_1 & \lambda_1 \epsilon & 0 & \cdots & 0 \\ & & & & \\ & & & & \end{bmatrix}$$

– claim –

$$= \left[\begin{array}{c|c} \begin{matrix} \lambda_1 \epsilon & & \\ & \lambda_1 \epsilon & \\ & & \ddots \end{matrix} & \\ \hline & \begin{matrix} \lambda_2 \epsilon & & \\ & \lambda_2 \epsilon & \\ & & \ddots \end{matrix} \end{array}\right]$$

We let the vector norm

$$\|x\|_* = \|(SD_\epsilon^{\cancel{1}})^{-1} x\|_\infty$$

Then

$$\|R\|_* = \max_x \frac{\|Rx\|_*}{\|x\|_*}$$

$$= \max_x \frac{\|(SD_\epsilon^{\cancel{1}})^{-1} R x\|_\infty}{\|(SD_\epsilon^{\cancel{1}})^{-1} x\|_\infty}. \text{ Let } y = (SD_\epsilon^{\cancel{1}})^{-1} x.$$

$$= \max_y \frac{\|(SD_\epsilon^{\cancel{1}})^{-1} R (SD_\epsilon) y\|_\infty}{\|y\|_\infty}$$

$$= \cancel{\max_y} \|(SD_\epsilon)^{-1} R (SD_\epsilon)\|_\infty$$

max row sum of the "ϵ Jordan form"

$$= \max |\lambda_i| + \epsilon = \rho(R) + \epsilon. \ \square$$

We can now say

Theorem. The iteration $x_{m+1} = Rx_m + c$ converges to solution of $Ax = b$ for all x_0 and all $b \iff \rho(R) < 1$.

Proof. If $\rho(R) \geq 1$, choose $x_0 - x$ to be an eigenvector with $\lambda \geq 1$. Then

$$(x_{m+1} - x) = R^{m+1}(x_0 - x) = \lambda^{m+1}(x_0 - x)$$

and this can't approach 0. If $\rho(R) < 1$, we can construct an operator norm so $\|R\| < 1$ as above and the iteration converges by Lemma. $\square$

Definition. The rate of convergence of $x_{m+1} = Rx_m + c$ is $r(R) := -\log_{10} \rho(R)$.

This is the number of additional correct digits in solution, per step.

To see this, we compute

$$(X_{m+1} - X) = R(X_m - X)$$

or

$$\| X_{m+1} - X \| \le \|R\| \, \|X_m - X\|$$

or

$$\frac{\| X_m - X \|}{\| X_{m+1} - X \|} \ge \frac{1}{\|R\|}$$

so

$$\log_{10} \| X_m - X \| - \log_{10} \| X_{m+1} - X \| \ge -\log_{10} \|R\|$$

But if this norm is chosen to be the $\| \ \|_*$ so $\|R\|_* \approx \rho(R) + \epsilon$, we get

$$\underbrace{\log_{10} \| X_m - X \|_*}_{-\#\text{ of correct digits in } X_m} \underbrace{- \log_{10} \| X_{m+1} - X \|_*}_{\#\text{ of correct digits in } X_{m+1}} \ge r(R) + O(\epsilon).$$

Next time: How to choose a splitting.