$$\vec{x}_{k+1} = \vec{c} + \overleftrightarrow{J} \cdot \vec{x}_k$$

$$= \vec{c} + \overleftrightarrow{J} \cdot (\vec{c} + \overleftrightarrow{J} \cdot \vec{x}_{k-1})$$

$$= \vec{c} + \overleftrightarrow{J} \cdot \vec{c} + \overleftrightarrow{J}^2 \cdot \vec{c} + \dots + \overleftrightarrow{J}^k \cdot \vec{c}$$

We assume now that $\overleftrightarrow{J}$ is written out in a coordinate system where it is diagonal:

$$\overleftrightarrow{J} = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_N \end{pmatrix}$$

Then, $$\vec{x}_{k+1} = \begin{pmatrix} \frac{1-\lambda_1^k}{1-\lambda_1} & & \\ & \ddots & \\ & & \frac{1-\lambda_N^k}{1-\lambda_N} \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_N \end{pmatrix}$$

Convergence rate:

$$\| \vec{x}_{k+1} - \vec{x}_k \| = \left[\sum_{i=1}^{N} |\lambda_i^k|^2 |C_i|^2 \right]^{1/2} = |\lambda_j^k| \, |C_j|$$

$\uparrow$ largest eigenvalue.

4/3.2

$$\|\vec{x}_k - \vec{x}\| = \left[\sum_{i=1}^{N} \frac{|\lambda_i^k|^2 |c_i|^2}{(1-\lambda_i|^2} \right]^{1/2}$$

$$\simeq \frac{|\lambda_j^k||c_j|}{|1-\lambda_j|}$$

$$\Rightarrow \quad \underline{\Lambda_j = |1-\lambda_j|}$$

where λ_j is the λ_i that has the largest absolute value.

Conjugate gradient algorithm

N^2 minimizations $\rightarrow$ N minimizations.

Construct two sequences of vectors g_i and h_i.

$$x_{i+1} = x_i + \lambda_i h_i$$

$$g_i = -\nabla f(x_i) = -A x_i + b$$

$$g_{i+1} = -\nabla f(x_{i+1}) = -\nabla f(x_i + \lambda_i h_i)$$

$$= -A(x_i + \lambda_i h_i) + b$$

$$= g_i - \lambda_i A h_i$$

Choose λ such that

$$\frac{d}{d\lambda_i} f(x_i + \lambda_i h_i) = h_i \nabla f(x_i + \lambda_i h_i)$$

$$= h_i(\lambda_i A h_i - g_i) = 0$$

$$\Rightarrow \quad \lambda_i = \frac{h_i g_i}{h_i A h_i}$$

But what h_i to use?

We choose it to be orthogonal to all preceding gradients g_i, i.e. orthogonal to the space spanned by the preceding g_i's.

In this way, we ensure that we will find the global minimum of f – the solution to the equation $Ax = b$ in N iterations.

Hence, $g_i \cdot h_j = 0$ for $i < j$.

If we set $$\gamma_i = \frac{g_{i+1} A h_i}{h_i A h_i}$$

and define

$$h_{i+1} = g_{i+1} + \gamma_i h_i,$$

we then have that

$$\boxed{\left.\begin{aligned} g_i \cdot g_j &= 0 \\ h_i A h_j &= 0 \\ g_i \cdot h_j &= 0 \end{aligned}\right\} \quad i \neq j}$$

We initiate by setting

$$h_0 = g_0 .$$

This algorithm converges in <u>N iterations.</u>

In practice, we set $N_{max} = 2N$ due to roundoff errors.

<u>Preconditioning</u>

General idea

$$Ax = b$$

Find C that is simple to invert.

Solve the following problem

$$\underbrace{(CAC^{-1})}_{A'}\underbrace{(Cx)}_{x'} = \underbrace{(Cb)}_{b'}$$

which is easier than the original problem (easier = faster convergence).

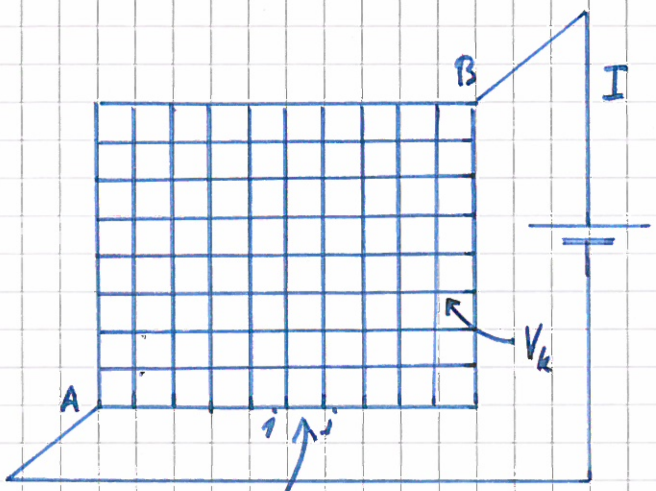
Then set $\underline{\underline{x = C^{-1}x'}}$

Example:

<u>Fourier acceleration</u>

Phys. Rev. Lett. <u>57</u>, 1336 (1986).

Concrete systems: Random resistor network.



g_{ij} - conductance that varies from link to link.

The Kirchhoff equations:

$$\sum_j g_{ij} \left(V_i - V_j \right) + I_i = 0$$

$$I_i = \begin{cases} 0 & \text{if } i \neq A, B \\ I & \text{if } i = A \text{ or } B \end{cases}$$

On matrix form:

$$GV + I = 0$$

Conductance matrix.

A very simple iterative process to solve this equation, is to set

$$\frac{dV}{dt} = GV + I .$$

Solution of this equation as $t \to \infty$:

$$\frac{dV}{dt} \to 0 \Rightarrow \underbrace{GV + I = 0}_{!}$$

Now, imagine all $g_{ij} = 1$ (they are in general not!)

$$\sum_j{}' g_{ij} (V_i - V_j) + I_i = 0$$

then turns into

$$\underbrace{\sum_j{}' (V_i - V_j)} + I_i = 0$$

This is the lowest-order finite difference approximation to $\nabla^2 V$.

Hence, we have

$$\underline{\frac{dV}{dt} = \nabla^2 V + I.}$$

We <u>Fourier transform</u> this equation:

$$\frac{d\tilde{V}}{dt} = -k^2 \tilde{V} + \tilde{I}.$$

$$\tilde{V}(\vec{k}) = \int d\vec{x}\, V(x)\, e^{i\vec{k}\cdot\vec{x}}.$$

Discretizing time (Euler integration)

$$\frac{\tilde{V}(\vec{k}, t+\epsilon) - \tilde{V}(\vec{k}, t)}{\epsilon} = -k^2 \tilde{V}(\vec{k}, t) + \tilde{I}(\vec{k})$$

$$\tilde{V}(\vec{k}, t+\epsilon) = \tilde{V}(\vec{k}, t) - \epsilon k^2 \tilde{V}(\vec{k}, t) + \epsilon \tilde{I}(\vec{k})$$

$$= (1 - \epsilon k^2) \tilde{V}(\vec{k}, t) + \epsilon \tilde{I}(\vec{k})$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$\tilde{V}(\vec{k}, t+n\epsilon) = (1 - \epsilon k^2)^n \tilde{V}(\vec{k}, t)$$

$$+ \left[(1 - \epsilon k^2)^{n-1} + (1 - \epsilon k^2)^{n-2} + \dots + 1 \right] \epsilon \tilde{I}(\vec{k})$$

$$= (1 - \epsilon k^2)^n \tilde{V}(\vec{k}, t)$$

$$+ \frac{1 - (1 - \epsilon k^2)^n}{1 - 1 + \epsilon k^2} \epsilon I(\vec{k})$$

$$\simeq e^{-\epsilon n k^2} \tilde{V}(\vec{k}, t)$$

$$+ \frac{1}{k^2} \left(1 - e^{-\epsilon n k^2}\right) I(\vec{k})$$

We now set $t = 0$ and $\tau = n\epsilon$:

$$\tilde{V}(\vec{k}, \tau) = e^{-\epsilon k^2 \tau}\, \tilde{V}(\vec{k}, 0) + \frac{1}{k^2}\left(1 - e^{-k^2\tau}\right) \tilde{I}(\vec{k})$$

Small k leads to slow evolution as $\tau \to \infty$.

This iterative method converges slowly because it is held back by the large-scale (i.e. small-k) structures in the problem.

Fourier acceleration.

Make ϵ k-dependent!

$$\boxed{\epsilon \to \epsilon(\vec{k})}$$

Set $\epsilon(\vec{k}) = \dfrac{\epsilon_0}{k^2}$

We then have

$$V(\vec{k}, \tau) = e^{-\epsilon(\vec{k})k^2\tau}\,\tilde{V}(\vec{k}, 0) + \frac{1}{k^2}\left(1 - e^{-\epsilon(\vec{k})k^2\tau}\right)\tilde{I}(\vec{k})$$

$$= e^{-\epsilon_0\tau}\,\tilde{V}(\vec{k}, 0) + \frac{1}{k^2}\left(1 - e^{-\epsilon_0\tau}\right)\tilde{I}(\vec{k})$$

All modes now converge equally fast.

This is how it is then implemented in practice:

$$\frac{dV}{dt} = GV + I$$

$\Rightarrow$

$$V_i(t+\epsilon) = V_i(t+\epsilon) + \epsilon \sum_j{}' g_{ij}(V_i(t) - V_j(t))$$
$$+ \epsilon I_i$$

We now make ϵ space dependent:

$$V_i(t+\epsilon) = V_i(t) + \sum_{ik}{}' \epsilon_{ik} \sum_j{}' g_{kj}(V_k(t) - V_j(t))$$
$$+ \sum_{ik}{}' \epsilon_{ik} I_k$$

We now Fourier transform –

$\mathcal{F}$ is the Fourier transform,

$\mathcal{F}^{-1}$ is its inverse.

$$\boxed{V_i(t+\epsilon) = V_i(t) + \epsilon_0 \sum_k{}' (\mathcal{F}^{-1} k^{-2} \mathcal{F})_{ik} \left(\sum_k{}' g_{kj} (V_k(t) - V_j(t)) + I_k \right).}$$

Notation trouble: t is no longer time, but an iteration parameter.

Singular value decomposition - SVD.

What if A cannot be inverted?

$$Ax = b$$

$M \times N$ matrix ; M equations

N unknowns.