Peeter Joot
peeter.joot@gmail.com

ECE1254H Modeling of Multiphysics Systems. Lecture 8: Conjugate gradient method. Taught by Prof. Piero Triverio

1.1 Recap: Summary of Gradient method

We'd like to solve

$$M\mathbf{x} = \mathbf{b}, \tag{1.1}$$

and introduce the residual between the application of any trial solution $\mathbf{y}$ to M and $\mathbf{b}$

$$\mathbf{r} = \mathbf{b} - M\mathbf{y}. \tag{1.2}$$

We seek to minimize the energy function

$$\Psi(\mathbf{y}) = \frac{1}{2}\mathbf{y}^\mathrm{T} M\mathbf{y} - \mathbf{y}^\mathrm{T}\mathbf{b}, \tag{1.3}$$

and follow the direction of steepest decent $\mathbf{r}^{(k)} = -\boldsymbol{\nabla}\Psi$ with the hope of finding the minimum of the energy function. That iteration is described by

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{r}^{(k)}, \tag{1.4}$$

and illustrated in fig. 1.1.

The problem of the Gradient method is that it introduces multiple paths as sketched in fig. 1.2. which lengthens the total distance that has to be traversed in the iteration.

1.2 Conjugate gradient method

The Conjugate gradient method makes the residual at step k orthogonal to all previous search directions

1. $\mathbf{r}^{(1)} \perp \mathbf{d}^{(0)}$
2. $\mathbf{r}^{(1)} \perp \mathbf{d}^{(0)}, \mathbf{d}^{(1)}$
3. $\cdots$

After n iterations, the residual will be zero.

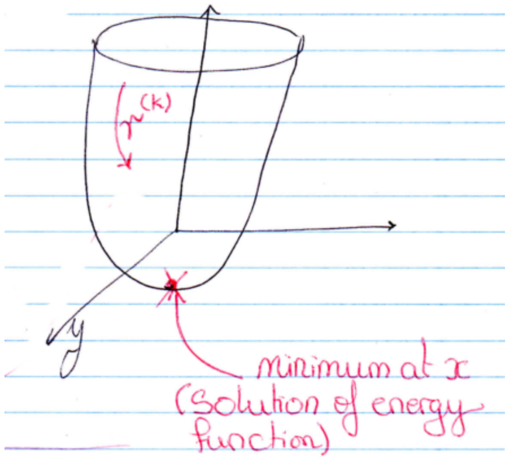

Figure 1.1: Gradient descent

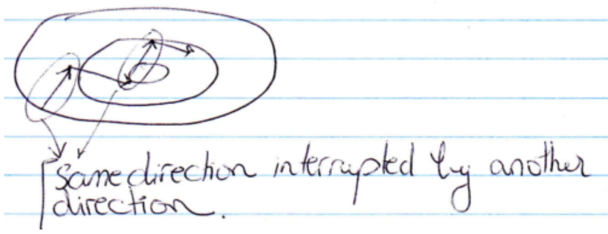

Figure 1.2: Gradient descent iteration

First iteration Given an initial guess $\mathbf{x}^{(0)}$, we proceed as in the gradient method

$$\mathbf{d}^{(0)} = -\boldsymbol{\nabla}\Psi(\mathbf{x}^{(0)}) = \mathbf{r}^{(0)}, \tag{1.5}$$

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha_0 \mathbf{r}^{(0)}, \tag{1.6}$$

with

$$\alpha_0 = \frac{\left(\mathbf{r}^{(0)}\right)^{\mathrm{T}} \mathbf{r}^{(0)}}{\left(\mathbf{r}^{(0)}\right)^{\mathrm{T}} M \mathbf{r}^{(0)}}, \tag{1.7}$$

so that the residual is

$$\begin{aligned} \mathbf{r}^{(1)} &= \mathbf{b} - M\mathbf{x}^{(1)} \\ &= \mathbf{b} - M\mathbf{x}^{(0)} - \alpha_0 M \mathbf{r}^{(0)} \\ &= \mathbf{b} - \alpha M \mathbf{r}^{(0)}. \end{aligned} \tag{1.8}$$

We want $\mathbf{r}^{(1)} \perp \mathbf{d}^{(0)}$.

Proof:

$$\begin{aligned} \langle \mathbf{d}^{(0)}, \mathbf{r}^{(1)} \rangle &= \langle \mathbf{d}^{(0)}, \mathbf{r}^{(0)} \rangle - \alpha_0 \langle \mathbf{d}^{(0)}, M\mathbf{r}^{(0)} \rangle \\ &= \langle \mathbf{d}^{(0)}, \mathbf{r}^{(0)} \rangle - \alpha_0 \langle \mathbf{r}^{(0)}, M\mathbf{r}^{(0)} \rangle. \end{aligned} \tag{1.9}$$

Second iteration

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \alpha_1 \mathbf{d}^{(1)}. \tag{1.10}$$

The conditions to satisfy are

$$\mathbf{d}^{(0)} \perp \mathbf{r}^{(2)} \tag{1.11a}$$

$$\mathbf{d}^{(1)} \perp \mathbf{r}^{(2)} \tag{1.11b}$$

We can confirm that condition of eq. (1.11a) is satisfied

$$\begin{aligned} \langle \mathbf{d}^{(0)}, \mathbf{r}^{(2)} \rangle &= \langle \mathbf{d}^{(0)}, \mathbf{b} - M\mathbf{x}^{(2)} \rangle \\ &= \langle \mathbf{d}^{(0)}, \mathbf{b} - M\mathbf{x}^{(1)} - \alpha_1 M \mathbf{d}^{(1)} \rangle \\ &\qquad\qquad = 0 \text{ because } \mathbf{d}^{(0)} \perp \mathbf{r}^{(1)} \\ &= \boxed{\langle \mathbf{d}^{(0)}, \mathbf{b} - M\mathbf{x}^{(1)} \rangle} - \alpha_1 \langle \mathbf{d}^{(0)}, \alpha_1 M \mathbf{d}^{(1)} \rangle \end{aligned} \tag{1.12}$$

This will be zero if we can impose an M orthogonality or conjugate zero condition

$$\langle \mathbf{d}^{(0)}, M\mathbf{d}^{(1)} \rangle = 0 \tag{1.13}$$

We impose eq. (1.13) to find a new search direction $\mathbf{d}^{(1)} = \mathbf{r}^{(1)} - \beta_0 \mathbf{d}^{(0)}$. The $\mathbf{r}^{(1)}$ is the term from the standard gradient method, and the $\beta_0 \mathbf{d}^{(0)}$ term is the conjugate gradient correction. This gives β_0

Table 1.1: LU vs Conjugate gradient order

	Full	Sparse
LU	$O(n^3)$	$O(n^{1.2-1.8})$
Conjugate gradient	$O(kn^2)$	5.4

$$\beta_0 = \frac{\langle \mathbf{d}^{(0)}, M\mathbf{r}^{(1)} \rangle}{\langle \mathbf{d}^{(0)}, M\mathbf{d}^{(0)} \rangle} \tag{1.14}$$

Imposing eq. (1.11b) gives us

$$\alpha_1 = \frac{\langle \mathbf{d}^{(0)}, \mathbf{r}^{(1)} \rangle}{\langle \mathbf{d}^{(1)}, M\mathbf{d}^{(1)} \rangle}. \tag{1.15}$$

Next iteration

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)} \\ \mathbf{d}^{(k)} &= \mathbf{r}^{(k)} - \beta_{k-1}\mathbf{d}^{(k-1)} \end{aligned} \tag{1.16}$$

The conditions to impose are

$$\begin{aligned} \mathbf{d}^{(0)} &\perp \mathbf{r}^{(k+1)} \\ &\vdots \\ \mathbf{d}^{(k-1)} &\perp \mathbf{r}^{(k+1)} \end{aligned} \tag{1.17}$$

However, we have only 2 degrees of freedom α, β, but have many conditions to impose. Impose the following to find β_{k-1}

$$\mathbf{d}^{(k-1)} \perp \mathbf{r}^{(k+1)} \tag{1.18}$$

(See slides)

1.3 Order analysis

Note that for $\mathbf{x}, \mathbf{y}$ in $\mathbb{R}^n$, $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T\mathbf{y}$ is $O(n)$, and $M\mathbf{x}$ is $O(n^2)$. Comparing to LU, where $k < n$

Final comments

1. How to select $\mathbf{x}^{(0)}$? We can get an initial rough estimate by solving a simplified version of the problem.
2. Do we have to save a memory space to store M ? No. If $\mathbf{y} = M\mathbf{z}$, we can calculate the product without physically storing the full matrix M.