Theoretical statement of CG

$\underline{x}_0 \longleftarrow$ Initial guess

$\underline{r}_0 \longleftarrow \underline{b} - A\underline{x}_0$

if $\|\underline{r}_0\| < \delta \Rightarrow$ TERMINATE

$\underline{p}_0 \longleftarrow \underline{r}_0$

$k \longleftarrow 0$

repeat

$\quad \alpha_k \longleftarrow \underline{r}_k^T \underline{r}_k / \underline{p}_k^T A \underline{p}_k$

$\quad \underline{x}_{k+1} \longleftarrow \underline{x}_k + \alpha_k \underline{p}_k$

$\quad \underline{r}_{k+1} \longleftarrow \underline{r}_k - \alpha_k A \underline{p}_k$

$\quad$ if $\|\underline{r}_{k+1}\| < \delta \Rightarrow$ TERMINATE

$\quad \beta_k \longleftarrow \underline{r}_{k+1}^T \underline{r}_{k+1} / \underline{r}_k^T \underline{r}_k$

$\quad \underline{p}_{k+1} \longleftarrow \underline{r}_{k+1} + \beta_k \underline{p}_k$

$\quad k \longleftarrow k+1$

Practical Implementation of CG

MatrixType A;

VectorType x, b, q, p, r;

$\rho^{old} \longleftarrow \infty$, $\underline{x} \longleftarrow$ initial guess (including constrained values)

$\underline{r} \longleftarrow \underline{b} - A\underline{x}$

Project ($\underline{r}$)

for $k = 0, 1, 2, \ldots$

$\quad$ if ($\|r\| < \delta$) $\Rightarrow$ TERMINATE ($\rightarrow \underline{x}$)

$\quad \rho = \underline{r}^T \underline{r}$

$\quad$ if ($k = 0$)

$\quad\quad \underline{p} \longleftarrow \underline{r}$

$\quad$ else

$\quad\quad \underline{p} \longleftarrow \underline{r} + \dfrac{\rho}{\rho_{old}} \cdot \underline{p}$

$\quad \underline{q} \longleftarrow A\underline{p}$

$\quad$ Project ($\underline{q}$)

$\quad \alpha \longleftarrow \rho / (\underline{p}^T \underline{q})$

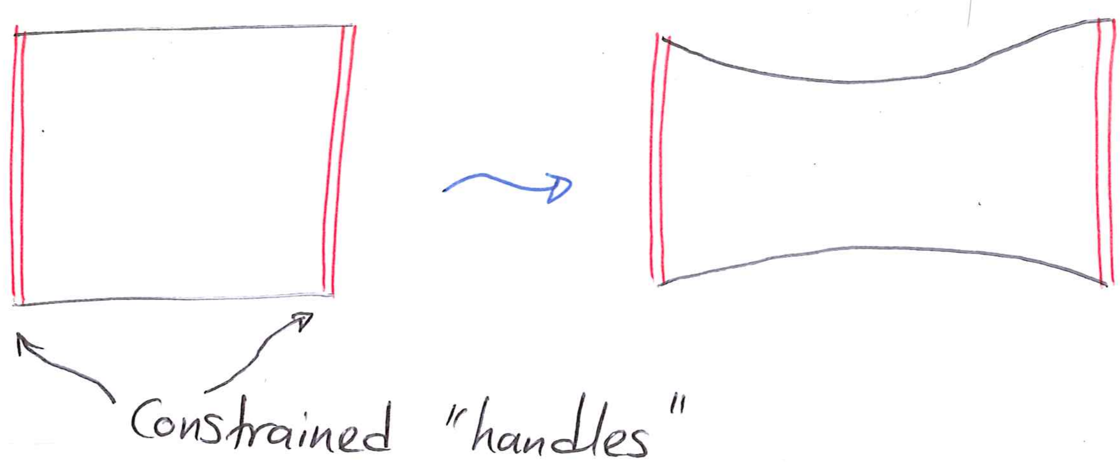
$\quad \underline{x} \longleftarrow \underline{x} + \alpha \underline{p}$

$\quad \underline{r} \longleftarrow \underline{r} - \alpha \underline{q}$

$\quad \rho_{old} \longleftarrow \rho$

Enforcing boundary conditions / constraints

Maybe some components of the solution vector $\underline{x}$ are already known, e.g. if parts of the mesh are directly moved as a user-imposed manipulation



Without loss of generality, assume that the particles (w/ values $\underline{x}$) are partitioned into free ($\underline{x}_F$) and constrained ($\underline{x}_C$). Then the system $A\underline{x} = \underline{b}$ is written in block form as:

$$\begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} \begin{pmatrix} \underline{x}_F \\ \underline{x}_C \end{pmatrix} = \begin{pmatrix} \underline{b}_F \\ \underline{b}_C \end{pmatrix}$$

If we are given externally imposed values $\underline{x}_C^*$ for the constrained particles, the bottom block becomes $\underline{x}_C = \underline{x}_C^*$

(Quiz: Why is it ok to just "throw away" the last block of equations, and replace them by $x_c = x_c^*$?)

In matrix terms, the system becomes:

$$\begin{pmatrix} A_{FF} & A_{FC} \\ 0 & I \end{pmatrix} \begin{pmatrix} \underline{x}_F \\ \underline{x}_c \end{pmatrix} = \begin{pmatrix} \underline{b}_F \\ \underline{x}_c^* \end{pmatrix}$$

This is a perfectly valid & solvable system, but just not with CG since we compromised the matrix symmetry!

Alternative: Rework the 1st block of equations as follows:

$$A_{FF}\,\underline{x}_F = \underline{b}_F - A_{FC}\,\underline{x}_c^*$$

$$\Rightarrow \underbrace{\begin{pmatrix} A_{FF} & 0 \\ 0 & 0 \end{pmatrix}}_{A^*} \begin{pmatrix} \underline{\delta}_F \\ \underline{\delta}_c \end{pmatrix} = \begin{pmatrix} b_F - A_{FF}\underline{\delta}_F - A_{FC}\underline{x}_c^* \\ 0 \end{pmatrix}$$

This matrix is symmetric

$$= \text{Project}\left[b - A \begin{pmatrix} \underline{x}_F^{(0)} \\ \underline{x}_c^* \end{pmatrix} \right]$$

(Define:

$$\text{Project}\begin{pmatrix} x_F \\ x_c \end{pmatrix} \rightarrow \begin{pmatrix} x_F \\ 0 \end{pmatrix}$$)

$$\underline{x}_F = \underline{x}_F^{(0)} + \underline{\delta}_F$$

Although we don't explicitly have matrix A^*, we can emulate the multiplication $A^* w$ as:

Multiply with A^* $[q \leftarrow A^* w]$

{

$w^* \leftarrow \text{Project}(w)$

$q^* \leftarrow A \cdot w^*$

$q \leftarrow \text{Project}(q^*)$

}

(which is all that CG requires, anyways)

The algorithm in page 1 results from eliminating "redundant" (repeated) projections.

Backward Euler as a correction

Assumptions:

- Linear Elasticity (for now)
- Presume initial guesses for x^{n+1} $(x^{n+1}_{(0)})$ and v^{n+1} $(v^{n+1}_{(0)})$
- We presume initial guesses satisfy B.E:

$$x^{n+1}_{(0)} = x^{n} + h\, v^{n+1}_{(0)} \qquad (1)$$

(e.g. use $v^{n+1}_{(0)} = v^{n}$, and $x^{n+1}_{(0)}$ from above)

- We seek corrections

$$\delta x := x^{n+1} - x^{n+1}_{(0)}$$

$$\delta v := v^{n+1} - v^{n+1}_{(0)}$$

$$\underline{x}^{n+1} = \underline{x}^{n} + h\,\underline{v}^{n+1} \overset{(1)}{\Longrightarrow} \boxed{\delta \underline{x} = h\,\delta \underline{v}}$$

$$\underline{v}^{n+1} = \underline{v}^{n} + hM^{-1}\left\{-K(x^{n+1} - X) - \gamma K v^{n+1}\right\}$$

$$\Rightarrow \delta \underline{v} = \underline{v}^{n} - \underline{v}^{n+1}_{(0)} + hM^{-1}\left\{-K(\delta \underline{x} + \underline{x}_{(0)} - X) - \gamma K(v^{n+1}_{(0)} + \delta \underline{v})\right\}$$

$\overset{(1)}{\Longrightarrow}$

$$\frac{1}{h}\delta \underline{x} = \underline{v}^{n} - \underline{v}^{n+1}_{(0)} + hM^{-1}\Big\{-K\delta x - \frac{\gamma}{h}K\delta\underline{x} + f_{el}(x^{n+1}_{(0)}) + f_{d}(v^{n+1}_{(0)})\Big\}$$

$$\Rightarrow \boxed{\left[\left(1+\frac{\gamma}{h}\right)K + \frac{1}{h^2}M\right]\delta\underline{x} = \frac{1}{h}M\left[\underline{v}^{n} - \underline{v}^{n+1}_{(0)}\right] + f_{el}(x^{n+1}_{(0)}) + f_{d}(v^{n+1}_{(0)})}$$