1 Combined approximation

The goal is to approximate $\mathbf{u}$ given $\mathbf{K}\mathbf{u} = \mathbf{f}$.

A reduced basis is generated using a previous factorization of $\mathbf{K}_0$. Let $\mathbf{K} = \mathbf{K}_0 + \Delta\mathbf{K}$, then the equilibrium equations can be written as

$$(\mathbf{K}_0 + \Delta\mathbf{K})\mathbf{u} = \mathbf{f}. \tag{1}$$

A recurrence equation can be found by moving terms to the right-hand side

$$\mathbf{K}_0\mathbf{u}_{k+1} = \mathbf{f} - \Delta\mathbf{K}\mathbf{u}_k. \tag{2}$$

Substitution gives $\mathbf{u}_{k+1}$ in terms of $\mathbf{u}_1 = \mathbf{K}_0^{-1}\mathbf{f}$

$$\mathbf{u}_{k+1} = \left(\boldsymbol{I} + \mathbf{B} + \mathbf{B^2} + \boldsymbol{...} + \mathbf{B}^{\boldsymbol{k}}\right)\mathbf{u}_1, \text{ where } \mathbf{B} = -\mathbf{K}_0^{-1}\Delta\mathbf{K}. \tag{3}$$

Thus, $\mathbf{u}$ can be expressed in the basis defined by

$$\begin{cases} \boldsymbol{\phi}_1 = \mathbf{K}_0^{-1}\mathbf{f} \\ \boldsymbol{\phi}_k = \mathbf{K}_0^{-1}\boldsymbol{\Delta}\mathbf{K}\phi_{k-1}. \end{cases}$$

If the change in the matrix $\mathbf{K}$ is small (given by $\Delta\mathbf{K}$), generating a small amount of basis vectors can yield an accurate approximation. Let $\mathbf{R_b}$ be the matrix consisting of the s first basis vectors, meaning $\mathbf{R_b} = (\boldsymbol{\phi_1},\ \boldsymbol{...},\ \boldsymbol{\phi_s})$. The approximation can be expressed as $\mathbf{R_b}\mathbf{y} = \tilde{\mathbf{u}}$. Substitution into the equilibrium equation, and premultiplying by $\mathbf{R_b}^T$ from the left gives a reduces system of equations

$$\mathbf{R_b}^T\mathbf{K}\mathbf{R_b}\mathbf{y} = \mathbf{R_b}^T\mathbf{f}. \tag{4}$$

Since the basis vectors $\boldsymbol{\phi}_k$ may be almost parallel, orthogonalizing them can make the scheme more numerically stable. Furthermore, orthogonalizing with respect to the matrix $\mathbf{K}$ (meaning $\hat{\boldsymbol{\phi}}_i^T\mathbf{K}\hat{\boldsymbol{\phi}}_j = \delta_{ij}$) makes the system very easy to solve. Let $\mathbf{V_b}$ denote the orthogonal basis generated by the Gram-Schmidt scheme. The resulting system is given by

$$\mathbf{V_b}^T\mathbf{K}\mathbf{V_b}\mathbf{z} = \mathbf{V_b}^T\mathbf{f}.$$

Due to the orthogonalization the left-hand side reduces to the identity matrix, $\mathbf{V_b}^T\mathbf{K}\mathbf{V_b} = \boldsymbol{I}$, and the approximation is given by

$$\tilde{\mathbf{u}} = \mathbf{V_b}\mathbf{z} = \mathbf{V_b}\mathbf{V_b}^T\mathbf{f}.$$

2 Nonlinear FEM

The displacements $\mathbf{a}$ are found by solving the nonlinear system of equations

$$\mathbf{r} = \boldsymbol{f}_{int}(\mathbf{z}, \mathbf{a}) - \boldsymbol{f}_{ext} = \mathbf{0}.$$

The stresses in the structure are given by the change in the strain energy with respect to the strain, or

$$\mathbf{S} = \frac{\partial w}{\partial \mathbf{E}}.$$

Discretizing the structure using finite elements, the internal forces in the structure are given by

$$\boldsymbol{f}_{int} = \int_\Omega \mathbf{B}^T\mathbf{S}dv.$$

A Newton-Raphson scheme is used to iteratively solve for the displacements $\mathbf{a}$,

$$\mathbf{K}\Delta\mathbf{a}_{k+1} = -\mathbf{r}_k$$

The displacements are then updated $\mathbf{a}_{k+1} = \mathbf{a}_k + \Delta\mathbf{a}_{k+1}$.

3 Line search

It can be shown that the number of iterations in the solution scheme can be reduced by introducing line search. Instead of updating the displacements directly using the update $\boldsymbol{a}^i = \boldsymbol{a}^{i-1} + \boldsymbol{\Delta a}$, the increment in displacements is used as a search direction to find the next displacements $\boldsymbol{a}^i = \boldsymbol{a}^{i-1} + \beta\boldsymbol{\Delta a}$. Introduce the quantity $r(\beta) = \boldsymbol{\Delta a}^T \boldsymbol{r}(\boldsymbol{a}^{i-1} + \beta\boldsymbol{\Delta a})$. The residual is minimized if β is chosen as

$$\hat{\beta} = \frac{r(0)}{r(0) + r(1)}.$$

Too large or small values of β may cause issues, so in practice β is chosen as

$$\beta = \max\Big(3,\ \min\Big(0.3,\ \hat{\beta}\Big)\Big).$$

4 Optimization

The end compliance can be used as a measure of the stiffness of the structure. If a displacement controlled scheme is used, maximizing the compliance results in a stiff structure.

$$g_o = c = \mathbf{u}_p^T \boldsymbol{f}_{int,p}.$$

The design variables $\boldsymbol{z}$ are filtered using a density filter to prevent a checkerboard pattern, meaning $\mathbf{z}_f = \mathbf{M}_f\mathbf{z}$. A SIMP interpolation scheme gives elasticity modulus

$$E_e(\rho_e) = E_{min} + (E_{max} - E_{min})\rho_e^q,$$

where q is usually chosen as 3.

A Neo-Hookean material model is used to model the structure, meaning the strain energy is given by

$$w^{NH} = \frac{\kappa}{2}\bigg(\frac{1}{2}(J^2 - 1) - ln(J)\bigg) + \frac{1}{2}\mu\bigg(J^{-2/3}tr(\mathbf{C}) - 3\bigg),$$

where $J = \det(\mathbf{F})$, $\mathbf{C}$ is the Cauchy-Green tensor $\mathbf{C} = \mathbf{F}^T\mathbf{F}$, $\kappa = \frac{E}{3(1-2\nu)}$ is the bulk modulus, and $\mu = \frac{E}{2(1+\nu)}$ is the shear modulus.

The strain energy is linear in the elasticity modulus $E_e(\rho_e)$, which is independent on the strain $\boldsymbol{E}$. This gives a very easy connection between the design variables and the internal forces, which can be evaluated choosing E = 1, and then multiplying by the interpolated elasticity modulus.

$$(\boldsymbol{f}_{int})_e = E_e(\rho_e)\int_{V_e} \mathbf{B}^T \boldsymbol{S}_e^{NH}\big|_{E=1} dV_e.$$

The sensitivity of the forces with respect to the element density is thus

$$\frac{d\boldsymbol{f}_{int,e}}{d\rho_e} = \frac{dE_e(\rho_e)}{d\rho_e}\int_{V_e} \mathbf{B}^T \boldsymbol{S}_e^{NH}\big|_{E=1} dV_e.$$

4.1 Sensitivities

Instead of computing the sensitivity of the goal function, an augmented goal function is analysed in order to reduce computational cost. Thus, consider the function

$$\tilde{g_0} = \mathbf{u}_p^T \boldsymbol{f}_{int,p} + \boldsymbol{\lambda}^T \boldsymbol{f}_{int,f}.$$

Assuming the structure is in equilibrium, this augmented goal function is identical to the original goal function. The sensitivity is given by

$$\begin{aligned}\frac{d\tilde{g_0}}{dz_e} &= \mathbf{u}_p^T\left(\frac{\partial \boldsymbol{f}_{int,p}}{\partial z_e} + \frac{\partial \boldsymbol{f}_{int,p}}{\partial \mathbf{u}_f}\frac{\partial \mathbf{u}_f}{\partial z_e}\right) + \boldsymbol{\lambda}^T\left(\frac{d\boldsymbol{f}_{int,f}}{dz_e} + \frac{d\boldsymbol{f}_{int,f}}{d\mathbf{u}_f}\frac{d\mathbf{u}_f}{dz_e}\right) = \\ &= \mathbf{u}_p^T\frac{\partial \boldsymbol{f}_{int,p}}{\partial z_e} + \boldsymbol{\lambda}^T\frac{\partial \boldsymbol{f}_{int,p}}{\partial z_e} + \left(\boldsymbol{\lambda}^T\frac{d\boldsymbol{f}_{int,f}}{d\mathbf{u}_p} + \mathbf{u}_p^T\frac{\partial \boldsymbol{f}_{int,p}}{\partial \mathbf{u}_f}\right)\frac{d\mathbf{u}_f}{dz_e}.\end{aligned}$$

The derivative $\frac{d\mathbf{u}_f}{dz_e}$ can be eliminated by asserting

$$\boldsymbol{\lambda}^T \frac{d\boldsymbol{f}_{int,f}}{d\mathbf{u}_p} + \mathbf{u}_p \frac{\partial \boldsymbol{f}_{int,p}}{\partial \mathbf{u}_f} = \mathbf{0}$$

Using the notation $\frac{d\boldsymbol{f}_{int,f}}{d\mathbf{u}_f} = \boldsymbol{K}_{ff}$, and $\frac{d\boldsymbol{f}_{int,p}}{d\mathbf{u}_f} = \boldsymbol{K}_{fp}$ we have

$$\begin{aligned} \frac{d\tilde{g_0}}{dz_e} &= \frac{d\boldsymbol{f}_{int,f}^T}{dz_e}\boldsymbol{\lambda} \\ \boldsymbol{K}_{ff}\boldsymbol{\lambda} &= -\boldsymbol{K}_{fp}\mathbf{u}_p. \end{aligned} \tag{5}$$

5 Reusing information

The idea of reusing information between optimization steps is what *Combined Approximation* is based on. *Combined Approximation* reuses information of the stiffness matrix from the previous step to speed up the solution of the non-linear system of residual equations, which is the most costly step in the optimization scheme as a whole. However, the change in design variables can also be used to find an initial guess to the increment in displacement. Expanding the residual equations in $\boldsymbol{z}$ and $\boldsymbol{u}$ we have

$$\boldsymbol{r}_f(\boldsymbol{z}^i, \boldsymbol{u}^i) \approx \boldsymbol{r}_f(\boldsymbol{z}^{i-1}, \boldsymbol{u}^{i-1}) + \frac{\partial \boldsymbol{r}_f^{i-1}}{\partial \boldsymbol{z}}\boldsymbol{\Delta z} + \frac{\partial \boldsymbol{r}_f^{i-1}}{\partial \boldsymbol{u}_f}\boldsymbol{\Delta u}_f.$$

Assuming the residual equations are solved at the previous step $\boldsymbol{r}^{i-1} = \mathbf{0}$, and asserting the residual equations are solved at the current step $\boldsymbol{r}^i = \mathbf{0}$, an initial guess for the displacements can be found

$$\frac{\partial \boldsymbol{r}_f^{i-1}}{\partial \boldsymbol{u}_f}\boldsymbol{\Delta u}_f = -\frac{\partial \boldsymbol{r}_f^{i-1}}{\partial \boldsymbol{z}}\boldsymbol{\Delta z}.$$

Since the value of $\frac{\partial \boldsymbol{r}_f^{i-1}}{\partial \boldsymbol{z}}$ was computed in the evaluation of the sensitivities in the previous step, and the matrix $\frac{\partial \boldsymbol{r}_f^{i-1}}{\partial \boldsymbol{u}_f}$ has already been factorized and used in the iterative scheme, (or CA can be used to find a good approximation) this is very cheap to evaluate. As we have discarded higher order terms, the accuracy depends on higher order derivatives and the the norm of $\boldsymbol{z}$, so in practice this initial guess of the displacement increment is only made if the norm of the change in design is small enough.

6 Pseudocode

```
Optimization scheme

for i = 1 to maxits
    - Solve for displacements and forces using NR
        if i == 1
            - take $n$ steps
        else
            - take 1 step with initial guess $\boldsymbol{u}_{i-1}^n$, and $\boldsymbol{K}_{i-1}^n$
        end if
    - Compute sensitivities, solving equation 5 with CA
    - Solve mma-subproblem
end for
```

Figure 1: The complete optimzation scheme for the nonlinear optimization.

```
Combined Approximation

- Initialize quantities
- $\boldsymbol{r}_1 = \mathbf{K}_0^{-1}\mathbf{f},\ \Delta\mathbf{K} = \boldsymbol{K} - \boldsymbol{K}_0$
- $\boldsymbol{v}_1 = (\boldsymbol{r}_1^T\mathbf{K}\boldsymbol{r}_1)^{-1/2}\boldsymbol{r}_1$
  for k = 2 to s
      - compute basis $\boldsymbol{r}_k = -\mathbf{K}_0^{-1}\Delta\mathbf{K}\boldsymbol{r}_{k-1}$
      - initialize orthogonalization $\boldsymbol{v}_k = \boldsymbol{r}_k$
        for j = 1 to k - 1
            - $\boldsymbol{v}_k -= (\boldsymbol{r}_k^T\mathbf{K}\boldsymbol{v}_j)\boldsymbol{v}_j$
        end for
      - normalize $\boldsymbol{v}_k *= (\boldsymbol{v}_k^T\mathbf{K}\boldsymbol{v}_k)^{-1/2}$
  end for
- Solve $\boldsymbol{u} = \boldsymbol{V}(\boldsymbol{V}^T\boldsymbol{f})$
```

Figure 2: Pseudocode for the combined approximation. It's very simple.

```
Newton-Raphson (Displacement Controlled)
Given design $\boldsymbol{z}_i = \boldsymbol{z}_{i-1} + \boldsymbol{\Delta z}$ at optimization step $i$:

    - Initialize quantities

      for k = 1 to # of loadsteps

          - Update displacement vector $\boldsymbol{u}_i^k = \boldsymbol{u}_i^{k-1} + \boldsymbol{\Delta u}$
            do while $||\boldsymbol{r}_{free}|| > r_{tol}$
                - Compute stiffness matrix $\boldsymbol{K}(\boldsymbol{u}_i^k, \boldsymbol{z}_i) = \boldsymbol{K}_i^k$
                  if $\cos(\theta(\boldsymbol{z}_i, \boldsymbol{z}_{i-1})) > c_{tol}$
                      - fetch $\boldsymbol{K}_{i-1}^k$, and $\boldsymbol{R}_{i-1}^k$
                      - solve $\boldsymbol{K}_i^k \boldsymbol{s} = -\boldsymbol{r}$ with CA
                  else
                      - factorize $\boldsymbol{K}_i^k = \left(\boldsymbol{R}_i^k\right)^T \boldsymbol{R}_i^k$
                      - solve $\boldsymbol{s} = -\boldsymbol{R}_i^k \Big\backslash \left(\boldsymbol{R}_i^k\right)^T \Big\backslash \boldsymbol{r}$
                      - save $\boldsymbol{R}_i^k$
                  end if
                - Search for $\boldsymbol{u}_i^k$ along the direction $\boldsymbol{s}$
                - Update residual $\boldsymbol{r} = \boldsymbol{r}(\boldsymbol{u}_i^k, \boldsymbol{z}_i)$
            end while
          - Save $\boldsymbol{K}_i^k$ and $\boldsymbol{u}_i^k$.

      end for
```

Figure 3: Newton Raphson scheme using the combined approximation for solving the linear equation system.

Both changes is design and changes in the displacements perturb $\boldsymbol{K}$ and can result in poor performance for CA if the perturbation in $\boldsymbol{K}$ is large. A check for the change in displacements can be used in tandem with the design change check to ensure CA performs well (although I have not found time to test such an implementation).