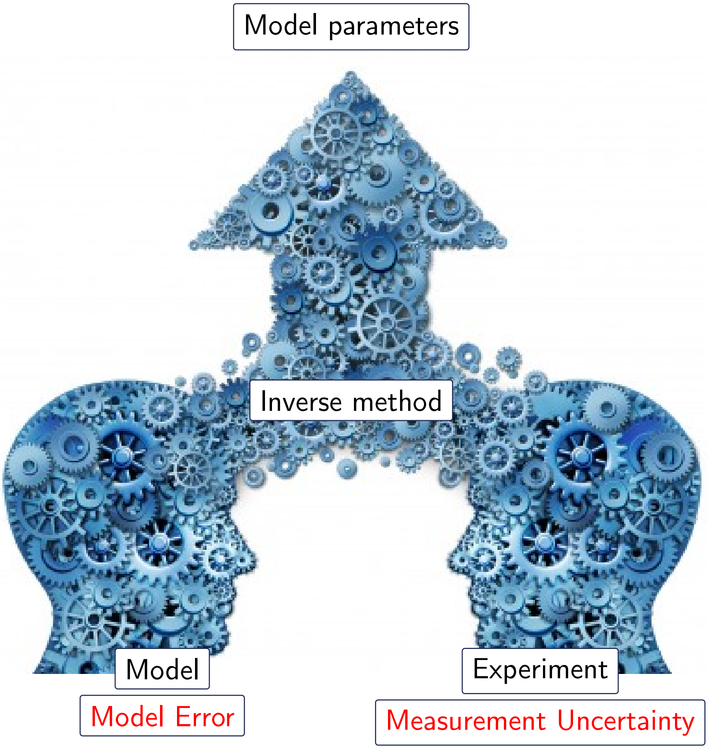
Inverse methods, the experimental - numerical dialog

Jan Neggers

What is a an inverse method



Defining the distance

Goals

- Use all available data,
- Consider uncertainty
- Weight each data source

Consider N observables

$$x_i = \mathcal{G}_i(\{p\})$$

Measurement with noise (variance γ_i^2)

$$\hat{x}_i = x_i + \zeta_i$$

Probability of one observable being x_i

$$P_i = \frac{1}{(2\pi)^{1/2}\gamma_i} \exp\left(-\frac{(\hat{x}_i - x_i)^2}{2\gamma_i^2}\right),$$

Probability of N observables

$$P = \frac{1}{(2\pi)^{N/2}\prod_{i=1}^{N}\gamma_i} \exp\left(-\sum_{i=1}^{N}\frac{(\hat{x}_i - x_i)^2}{2\gamma_i^2}\right)$$

Log-Likelihood

$$\eta^2 = \sum_{i=1}^{N}\frac{(\hat{x}_i - x_i)^2}{\gamma_i^2}$$

Different observables $N = N_1 + N_2$

$$\eta^2 = \frac{1}{\gamma_1^2}\sum_{i=1}^{N_1}(\hat{x}_i - x_i)^2 + \frac{1}{\gamma_2^2}\sum_{j=N_1+1}^{N}(\hat{x}_j - x_j)^2$$

Measurement aggregation

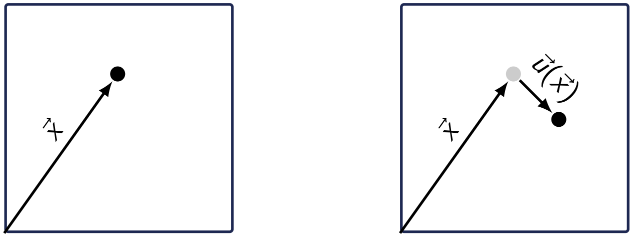
$$\eta^2 = \eta_1^2 + \eta_2^2$$

Weight factors

$$\omega_i = \frac{1}{\gamma_i^2}$$

Digital Image Correlation

reference image f deformed image g



Brightness Conservation

$$f(\vec{x}) \approx g(\vec{x} + \vec{u}(\vec{x}, \{a\}))$$

Least Squares cost function

$$\eta^2 = \frac{1}{2\gamma^2} \sum_{k=1}^{N_k} \Big(f(\vec{x}_k) - g(\vec{x}_k + \vec{u}(\vec{x}_k, \{a\})) \Big)^2$$

Basis functions

$$\vec{\varphi}_i = \frac{\partial \vec{u}}{\partial a_i}$$

Iterative Optimization

$$\{a\}^{(\mathrm{it}+1)} = \{a\}^{\mathrm{it}} + \{\delta a\}$$

$$[M]\{\delta a\} = \{b\}$$

$$M_{ij} = \frac{1}{2\gamma^2} \sum_{k=1}^{N_k} \vec{\nabla} f_k \vec{\varphi}_{ki} \vec{\varphi}_{kj} \vec{\nabla} f_k$$

$$b_i = \frac{1}{2\gamma^2} \sum_{k=1}^{N_k} \vec{\nabla} f_k \vec{\varphi}_{ki} r_k$$

Global DIC:
$\vec{\varphi} \rightarrow$ FE-shapefunctions
$\{a\} \rightarrow$ nodes

Implementing DIC

$$M_{ij} = \frac{1}{2\gamma^2} \sum_{k=1}^{N_k} \vec{\nabla} f_k \vec{\varphi}_{ki} \vec{\varphi}_{kj} \vec{\nabla} f_k$$

$$b_i = \frac{1}{2\gamma^2} \sum_{k=1}^{N_k} \vec{\nabla} f_k \vec{\varphi}_{ki} r_k$$

$$r_k = f(\vec{x}_k) - \tilde{g}(\vec{x}_k)$$

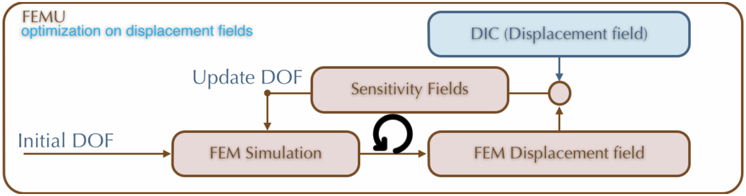
$$\tilde{g} = g(\vec{x}_k + \vec{u}(\vec{x}_k, \{a\})$$

$$[M]\{\delta a\} = \{b\}$$

$$\{a\}^{(\mathrm{it}+1)} = \{a\}^{\mathrm{it}} + \{\delta a\}$$

```
% image gradient
[fx, fy] = gradient(f);

% shapefunctions
phi = TriangleShapefunGridded(coor, conn,
     size(f));

% back-deformed image
gt = interp2(X,Y,g,X+Ux,Y+Uy)

% residual
r = f - gt;

% The shapefunction gradient image product
Lx = phi .* repmat( fx(:), 1, Nn);
Ly = phi .* repmat( fy(:), 1, Nn);

% Hessian
M = zeros( 2*Nn, 2*Nn );
M(Ix,Ix) = transpose(Lx) * Lx;
M(Ix,Iy) = transpose(Lx) * Ly;
M(Iy,Ix) = transpose(Ly) * Lx;
M(Iy,Iy) = transpose(Ly) * Ly;

% the Jacobian matrix
b = zeros(2*Nn,1);
b(Ix,1) = transpose(Lx) * res(:);
b(Iy,1) = transpose(Ly) * res(:);

% solve the system
da = M \ b;

% update the dof
a = a + da;
```

Finite Element Method Updating



FEMU-U displacement cost function

$$\eta_U^2(\{p\}) = \frac{1}{2\gamma_U^2} \sum_{t=1}^{N_t} \sum_{k=1}^{N_k} \left(U_{tk}^{\text{exp}} - U_{tk}(\{p\}) \right)^2$$

FEMU-F cost function

$$\eta_F^2(\{p\}) = \frac{1}{\gamma_F^2} \sum_{t=1}^{N_t} \left(F_t^{\text{exp}} - F_t(\{p\}) \right)^2$$

FEMU-UF cost function

$$\eta_e^2 = \eta_I^2 + \eta_F^2$$

FEMU-U system of equations

$$\begin{aligned} [H_U]\{\delta p\} &= \{J_U\} \\ [H_U]_{ij} &= [S_U]_i [S_U]_j \\ [J_U]_i &= [S_U]_i \{R_U\} \end{aligned}$$

FEMU-F system of equations

$$\begin{aligned} [H_F]\{\delta p\} &= \{J_F\} \\ [H_F]_{ij} &= [S_F]_i [S_F]_j \\ [J_F]_i &= [S_F]_i \{R_F\} \end{aligned}$$

FEMU-UF system of equations

$$([H_U] + [H_F]) \{\delta p\} = \{J_U\} + \{J_F\}$$

Implementing FEMU

$$[H_U]\{\delta p\} = \{J_U\}$$
$$[H_U]_{ij} = [S_U]_i[S_U]_j$$
$$[J_U]_i = [S_U]_i\{R_U\}$$

$$[H_F]\{\delta p\} = \{J_F\}$$
$$[H_F]_{ij} = [S_F]_i[S_F]_j$$
$$[J_F]_i = [S_F]_i\{R_F\}$$

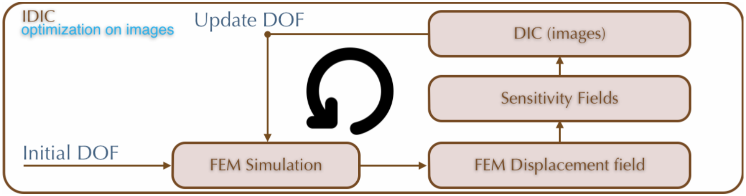
$$[S_U]_i = \frac{\partial U_{tk}(\{p\})}{\partial p_i} \approx \frac{U_{tk}(\{p\}^{\text{per}}) - U_{tk}(\{p\}^{\text{ref}})}{p_i^{\text{per}} - p_i^{\text{ref}}}$$
$$[S_F]_i = \frac{\partial F_t(\{p\})}{\partial p_i} \approx \frac{F_{tk}(\{p\}^{\text{per}}) - F_{tk}(\{p\}^{\text{ref}})}{p_i^{\text{per}} - p_i^{\text{ref}}}$$
$$p_i^{\text{per}} = p_i^{\text{ref}} + \epsilon p_i^{\text{ref}}$$

Relative sensitivity

$$[S_U]_i^* \approx \frac{U_{tk}(\{p\}^{\text{per}}) - U_{tk}(\{p\}^{\text{ref}})}{\epsilon} = [S_U]_i p_i^{\text{ref}}$$
$$[S_F]_i^* \approx \frac{F_{tk}(\{p\}^{\text{per}}) - F_{tk}(\{p\}^{\text{ref}})}{\epsilon} = [S_F]_i p_i^{\text{ref}}$$

```
% run abaqus
[Uref, Fref, Su, Sf] = abaqus(p, ABQ);

% compute the residual
Ru = Uexp - Uref;
Rf = Fexp - Fref;

% compute the Hessian
Hu = transpose(Su) * Su;
Hf = transpose(Sf) * Sf;

% compute the Jacobian matrix
Ju = transpose(Su) * Ru;
Jf = transpose(Sf) * Rf;

% combine the two systems
H = Wu * Hu + Wf * Hf;
J = Wu * Ju + Wf * Jf;

% solve for the update in the parametesr:
dp = H \ J;

% update the parameters
p = p + dp .* p;
```

Integrated DIC



I-DIC cost function

$$\eta_I^2(\{p\}) = \frac{1}{2\gamma_I^2} \sum_{t=1}^{N_t} \sum_{k=1}^{N_k} \left(f_k - \tilde{g}_{kt}(\{p\}) \right)^2$$

Force cost function

$$\eta_F^2(\{p\}) = \frac{1}{\gamma_F^2} \sum_{t=1}^{N_t} \left(F_t^{\exp} - F_t(\{p\}) \right)^2$$

Single experiment cost function

$$\eta_e^2 = \eta_I^2 + \eta_F^2$$

IDIC system of equations

$$[H_I]\{\delta p\} = \{J_I\}$$
$$[H_I]_{ij} = [S_U]_i \{\vec{\nabla} f\} [\vec{\varphi}]^T [\vec{\varphi}] \{\vec{\nabla} f\} [S_I]_j$$
$$[J_I]_i = [S_U]_i \{\vec{\nabla} f\} [\vec{\varphi}]^T \{r\}$$

Force system of equations

$$[H_F]\{\delta p\} = \{J_F\}$$
$$[H_F]_{ij} = [S_F]_i [S_F]_j$$
$$[J_F]_i = [S_F]_i \{R_F\}$$

Total system of equations

$$([H_I] + [H_F])\{\delta p\} = \{J_U\} + \{J_F\}$$

Implementing IDIC

$$
\begin{aligned}
[H_I]\{\delta p\} &= \{J_I\} \\
[H_I]_{ij} &= [S_U]_i\{\vec{\nabla} f\}[\vec{\varphi}]^T[\vec{\varphi}]\{\vec{\nabla} f\}[S_I]_j \\
[J_I]_i &= [S_U]_i\{\vec{\nabla} f\}[\vec{\varphi}]^T\{r\} \\
&\approx [\vec{S_U}]_i[M]\delta\{a\} \rightarrow \text{FEMU}
\end{aligned}
$$

```
% compute the Hessian
Hu = transpose(Su) * M * Su;

% compute the Jacobian Matrix
Ju = transpose(Su) * b;
```

What

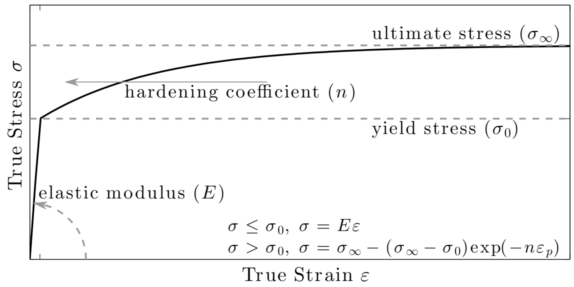
- Mechanical filter $\rightarrow$ experimental noise
- Parameter extractor $\rightarrow$ unknown parameters
- Rich overlay $\rightarrow$ internal variables

How

- Model an experiment
- Inverse method
- Dialog with experimental data
- Synchronization

Example Case - Model

- Aluminum Alloy 2219
- 2 Elastic parameters
  E, ν
- 3 Isotropic Exponential Hardening parameters
  σ_0, σ_∞, n



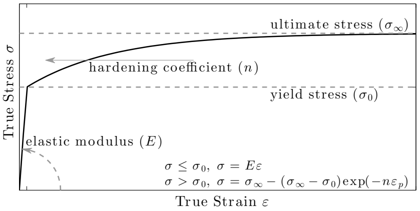
Voce hardening

$$
\begin{aligned}
\sigma &= E\varepsilon, && \sigma \le \sigma_0 \\
\sigma &= \sigma_\infty - (\sigma_\infty - \sigma_0)\exp(-n\varepsilon_p), && \sigma > \sigma_0
\end{aligned}
$$

[1] E. Voce, A Practical Strain Hardening Function, Metallurgia. (1955)

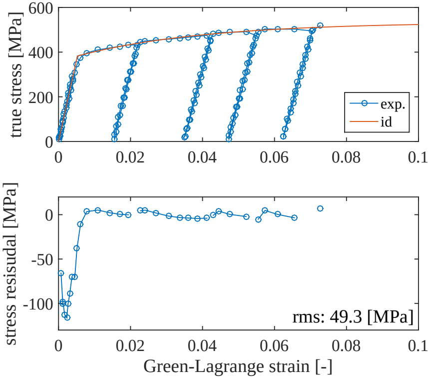
Stress-strain identification

- Extract Stress and Strain
- $DIC \rightarrow \varepsilon$
- $\sigma = F/A = F(1+\varepsilon)/A_0$
- 49 MPa stress residual



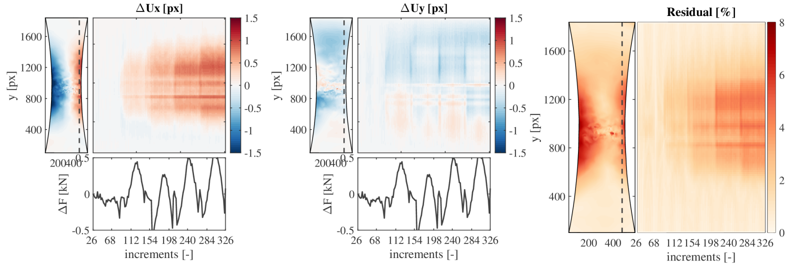
Parameters

p_1	p_2	p_3	p_4	p_5
E	ν	σ_0	σ_∞	n
[GPa]	[-]	[MPa]	[MPa]	[-]
73.2	0.3*	382	534	28.2

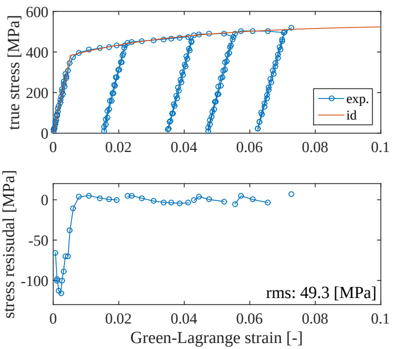


- Is this a good identification?

Stress-strain identification - Residuals



- Is this a good identification?



Exercises

three exercises:

1. DIC
2. FEMU
3. IDIC

Exercise 01 - DIC

Easy

1. inspect the image series, what type of experiment is this?
2. what data type are the images after reading?
3. why convert them to floating point?
4. how is the DIC initialized? Does it matter? Can you think of a smarter choice?

Normal

1. what is the size of the support of a node?
2. each nodes has two degrees of freedom, how are they arranged in M, a and b?
3. how are the index vectors I_x and I_y used to achieve this?
4. is the DIC solution dependent on the initialization?
5. what influences the DIC accuracy?

Hard

1. are the shapefunctions φ vector fields or scalar fields?
2. how would you define the spatial resolution of GDIC?
3. is there a relation between M and the accuracy of the solution?

Exercise 02 - FEMU

Easy

1. what do the five parameters in $\{p\}$ represent?
2. what do the sensitivity fields represent?

Normal

1. are each of the parameters equally sensitive? How can you tell?
2. look at the help of abaqus.m, it discusses relative sensitivity fields. What is the consequence?
3. what is the relative weight of each of the nodes in the cost function?
4. the Levenberg-Marquardt implementation slows the convergences but improves robustness, try some different LM strengths. Which parameter becomes unstable first?

Hard

1. plot the 5 sensitivity fields, do you understand what they mean?
2. what are the eigenvectors and eigenvalues of H, do you understand what they mean?
3. improve the code by proposing a correction for the weight of each node
4. explain how the Levenberg-Marquardt works, what is the added cost function?

Exercise 03 - IDIC

Easy

1. run both the FEMU version of the code and the IDIC version of the code, how are they different?

Normal

1. this time the displacement fields are scaled with the pixelsize, why?
2. similarly, we use the gray value uncertainty γ_i to weight each cost function, why?
3. under which conditions would IDIC and FEMU-M be equal? and when would they be different
4. after having calibrated our model, what does the digital twin allow you to do?

Hard

1. $[S_U]^T[M][S_U]$ is easier to compute in a loop over the increments, why?
2. explain why the two Jacobian matrices $\{J_U\}$ (FEMU-M and IDIC) are approximate each other.