Numerical Jacobian

Given a system of non-linear algebraic equations, $\underline{r}(\underline{u}) = \underline{0}$, the most difficult step in applying the Newton-Raphson method is forming the Jacobian matrix $\underline{\underline{J}} = \frac{d\underline{r}}{d\underline{u}}$

$$J_{ij} = \frac{\partial r_i}{\partial u_j} = \begin{pmatrix} \frac{\partial r_1}{\partial u_1} & \frac{\partial r_1}{\partial u_2} & \cdots & \frac{\partial r_1}{\partial u_N} \\ \frac{\partial r_2}{\partial u_1} & \frac{\partial r_2}{\partial u_2} & & \frac{\partial r_2}{\partial u_N} \\ \frac{\partial r_3}{\partial u_1} & \frac{\partial r_3}{\partial u_2} & & \frac{\partial r_3}{\partial u_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial r_N}{\partial u_1} & \frac{\partial r_N}{\partial u_2} & & \frac{\partial r_N}{\partial u_N} \end{pmatrix}$$

$$\Rightarrow \quad \underline{\underline{J}} = \begin{bmatrix} \frac{d\underline{r}}{du_1} & \frac{d\underline{r}}{du_2} & \frac{d\underline{r}}{du_3} & \cdots & \frac{d\underline{r}}{du_4} \end{bmatrix}$$

The columns of $\underline{\underline{J}}$ can be created by perturbing one unknown at a time and compute the derivative with finite differences.

General code for numerical Jacobian

residual ↓ unknowns ↓ ↓ perturbation

$[J]$ = comp_jacobian ($\underline{r}$, $\underline{u}$, ε)

↑ anonymous function

```
n = length(u)

u_perturb = u;

for i = 1 : n

    u_perturb(i) = u_perturb(i) + eps;

    J(:,i) = ( r(u_perturb) - r(u) ) / ε;

    u_perturb(i) = u(i);

end
```

This approach works for any residual function. but it is generally expensive.

If $\underline{r}(\underline{y})$ is complicated it can be a separate function.

General outline

```
for n = 1 : N        % time-stepping loop
    θold = θ         % save old sol'n   θold = θ^n
    while nres > tol || ndθ > tol || k < kmax
        dθ = -J(θ) \ r(θ, θold)
        θ = θ + dθ
    end
end
```