Deep learned one-iteration nonlinear solver for solid mechanics

Tan N. Nguyen, Jaehong Lee, Liem Dinh-Tien, Minh Dang

July 18, 2023

Outline

Abstract

Nowadays, there are a lot of iterative algorithms which have been proposed for nonlinear problems of solid mechanics. The existing biggest drawback of iterative algorithms is the requirement of numerous iterations and computation to solve these problems. This can be found clearly when the large or complex problems with thousands or millions of degrees of freedom are solved. To overcome completely this difficulty, the novel one-iteration nonlinear solver (OINS) using time series prediction and the modified Riks method (M-R) is proposed in this paper.

Abstract

OINS is established upon the core idea as follows: (1) Firstly, we predict the load factor increment and the displacement vector increment and the convergent solution of the considering load step via the predictive networks which are trained by using the load factor and the displacement vector increments of the previous convergence steps and group method of data handling (GMDH);(2) Thanks to the predicted convergence solution of the load step is very close to or identical with the real one, the prediction phase used in any existing nonlinear solvers is eliminated completely in OINS. Next, the correction phase of the M-R is adopted and the OINS iteration is started at the predicted convergence point to reach the convergent solution. The training process and the applying process of GMDH are continuously conducted and repeated during the nonlinear analysis in order to predict the convergence point at the beginning of each load step. Through numerical investigations, we prove that OINS is powerful, highly accurate and only needs about one iteration per load step. Thus, OINS significantly saves number of iterations and a huge amount of computation compared with the conventional methods. Especially, OINS not only can detect limit, inflection, and other special points but also can predict exactly various types of instabilities of structures.

Outline

Background

Nowadays, there are a lot of iterative algorithms which have been proposed for nonlinear problems of solid mechanics. The existing biggest drawback of iterative algorithms is the requirement of numerous iterations and computation to solve these problems. This can be found clearly when the large or complex problems with thousands or millions of degrees of freedom are solved.

Nonlinear analysis of Isotropic Shells using First-order shell Deformation Theory (FSDT)

The study will evaluate the efficiency and reliability of OINS in solid mechanics via geometrically nonlinear analysis of shells using FSDT. The strain vectors using FSDT are expressed by:

$$\varepsilon = \{\varepsilon_{xx},\, \varepsilon_{yy},\, \gamma_{xy}\}^T = \varepsilon_0 + z\kappa_b$$
$$\gamma = \{\gamma_{xz},\, \gamma_{yz}\}^T = \varepsilon_s,$$

with

$$\varepsilon_0 = \varepsilon_L + \varepsilon_N;\ \varepsilon_L = \begin{Bmatrix} u_{0,x} + w_0/R \\ \nu_{0,y} \\ u_{0,y} + \nu_{0,x} \end{Bmatrix};\ \varepsilon_N = \frac{1}{2}\begin{Bmatrix} w_{0,x}^2 \\ w_{0,y}^2 \\ 2w_{0,xy} \end{Bmatrix}$$

$$\kappa_b = \begin{Bmatrix} \beta_{x,x} \\ \beta_{y,y} \\ \beta_{x,y} + \beta_{y,x} \end{Bmatrix};\ \varepsilon_s = \begin{Bmatrix} -u_0/R + w_{0,x} + \beta_x \\ w_{0,y} + \beta_y \end{Bmatrix}$$

Nonlinear analysis of Isotropic Shells using First-order shell Deformation Theory (FSDT)

The study will evaluate the efficiency and reliability of OINS in solid mechanics via geometrically nonlinear analysis of shells using FSDT. The strain vectors using FSDT are expressed by:

$$\begin{aligned}\varepsilon &= \{\varepsilon_{xx},\ \varepsilon_{yy},\ \gamma_{xy}\}^T = \varepsilon_0 + z\kappa_b \\ \gamma &= \{\gamma_{xz},\ \gamma_{yz}\}^T = \varepsilon_s,\end{aligned}$$

with

$$\varepsilon_0 = \varepsilon_L + \varepsilon_N;\ \varepsilon_L = \begin{Bmatrix} u_{0,x} + w_0/R \\ \nu_{0,y} \\ u_{0,y} + \nu_{0,x} \end{Bmatrix};\ \varepsilon_N = \frac{1}{2}\begin{Bmatrix} w_{0,x}^2 \\ w_{0,y}^2 \\ 2w_{0,xy} \end{Bmatrix}$$

$$\kappa_b = \begin{Bmatrix} \beta_{x,x} \\ \beta_{y,y} \\ \beta_{x,y} + \beta_{y,x} \end{Bmatrix};\ \varepsilon_s = \begin{Bmatrix} -u_0/R + w_{0,x} + \beta_x \\ w_{0,y} + \beta_y \end{Bmatrix}$$

Nonlinear analysis of Isotropic Shells using First-order shell Deformation Theory (FSDT)

and we denote Ω to be the initial configuration of the shell. At a specific application, i.e. focusing on the istropic shell of FSDT, NURBS basis function N_A is utilized. The study entitled "NURBS-based postbuckling analysis of functionally graded carbon nanotube-reinforced composite shells" thoroughly discussed the derivation of shell formulation based on NURBS.

FSDT on the Nonlinear Isogeometric Analysis of Shells

At the ith iteration and mth load increment, the system of incremental equation is expressed as

The FG-CNTRC Shell Formulation based on NURBS

$$K_T(q_m)\Delta^i q_m = {}^iF_{\text{ext},m} - {}^iF_{\text{int},m}$$

where

$$K_T = \int_\Omega \left[\begin{Bmatrix} B_A^L \\ B_A^b \\ B_A^s \end{Bmatrix} + \begin{Bmatrix} B_A^N \\ 0 \\ 0 \end{Bmatrix} \right]^T \begin{bmatrix} D^p & 0 & 0 \\ 0 & D^b & 0 \\ 0 & 0 & D^s \end{bmatrix} \left[\begin{Bmatrix} B_A^L \\ B_A^b \\ B_A^s \end{Bmatrix} + \begin{Bmatrix} B_A^N \\ 0 \\ 0 \end{Bmatrix} \right] d\Omega$$
$$+ \int_\Omega (B_A^g)^T \begin{bmatrix} N_x & N_{xy} \\ N_{xy} & N_y \end{bmatrix} B_A^g d\Omega$$

FSDT on the Nonlinear Isogeometric Analysis of Shells

At the ith iteration and mth load increment, the system of incremental equation is expressed as

The FG-CNTRC Shell Formulation based on NURBS

$$K_T(q_m)\Delta^i q_m = {}^iF_{\text{ext},m} - {}^iF_{\text{int},m}$$

where

$$K_T = \int_\Omega \left[\begin{Bmatrix} B_A^L \\ B_A^b \\ B_A^s \end{Bmatrix} + \begin{Bmatrix} B_A^N \\ 0 \\ 0 \end{Bmatrix} \right]^T \begin{bmatrix} D^p & 0 & 0 \\ 0 & D^b & 0 \\ 0 & 0 & D^s \end{bmatrix} \left[\begin{Bmatrix} B_A^L \\ B_A^b \\ B_A^s \end{Bmatrix} + \begin{Bmatrix} B_A^N \\ 0 \\ 0 \end{Bmatrix} \right] d\Omega$$

$$+ \int_\Omega (B_A^g)^T \begin{bmatrix} N_x & N_{xy} \\ N_{xy} & N_y \end{bmatrix} B_A^g d\Omega$$

FC-CNTRC based on NURBS Basis Function

Derivation of the equation is based on the related literature from the same author detailing the NURBS basis functions. We have

$$B_A^L = \begin{bmatrix} N_{A,x} & 0 & \frac{1}{R}N_A & 0 & 0 \\ 0 & N_{A,y} & 0 & 0 & 0 \\ N_{A,y} & N_{A,x} & 0 & 0 & 0 \end{bmatrix}; \quad B_A^b = \begin{bmatrix} 0 & 0 & 0 & N_{A,x} & 0 \\ 0 & 0 & 0 & 0 & N_{A,y} \\ 0 & 0 & 0 & N_{A,y} & N_{A,x} \end{bmatrix}$$

$$B_A^s = \begin{bmatrix} -\frac{1}{R}N_A & 0 & N_{A,x} & N_A & 0 \\ 0 & 0 & N_{A,y} & 0 & N_A \end{bmatrix}$$

and

$$B_A^N = AB_A^g = \begin{bmatrix} w_{0,x} & 0 \\ 0 & w_{0,y} \\ w_{0,y} & w_{0,x} \end{bmatrix} \begin{bmatrix} 0 & 0 & N_{A,x} & 0 & 0 \\ 0 & 0 & N_{A,y} & 0 & 0 \end{bmatrix}$$

where w_0 is the radial deflector of the panel.

FC-CNTRC based on NURBS Basis Function

Derivation of the equation is based on the related literature from the same author detailing the NURBS basis functions. We have

$$B_A^L = \begin{bmatrix} N_{A,x} & 0 & \frac{1}{R}N_A & 0 & 0 \\ 0 & N_{A,y} & 0 & 0 & 0 \\ N_{A,y} & N_{A,x} & 0 & 0 & 0 \end{bmatrix}; \quad B_A^b = \begin{bmatrix} 0 & 0 & 0 & N_{A,x} & 0 \\ 0 & 0 & 0 & 0 & N_{A,y} \\ 0 & 0 & 0 & N_{A,y} & N_{A,x} \end{bmatrix}$$

$$B_A^s = \begin{bmatrix} -\frac{1}{R}N_A & 0 & N_{A,x} & N_A & 0 \\ 0 & 0 & N_{A,y} & 0 & N_A \end{bmatrix}$$

and

$$B_A^N = AB_A^g = \begin{bmatrix} w_{0,x} & 0 \\ 0 & w_{0,y} \\ w_{0,y} & w_{0,x} \end{bmatrix} \begin{bmatrix} 0 & 0 & N_{A,x} & 0 & 0 \\ 0 & 0 & N_{A,y} & 0 & 0 \end{bmatrix}$$

where w_0 is the radial deflector of the panel.

FC-CNTRC based on NURBS Basis Function

Derivation of the equation is based on the related literature from the same author detailing the NURBS basis functions. We have

$$B_A^L = \begin{bmatrix} N_{A,x} & 0 & \frac{1}{R}N_A & 0 & 0 \\ 0 & N_{A,y} & 0 & 0 & 0 \\ N_{A,y} & N_{A,x} & 0 & 0 & 0 \end{bmatrix}; \quad B_A^b = \begin{bmatrix} 0 & 0 & 0 & N_{A,x} & 0 \\ 0 & 0 & 0 & 0 & N_{A,y} \\ 0 & 0 & 0 & N_{A,y} & N_{A,x} \end{bmatrix}$$

$$B_A^s = \begin{bmatrix} -\frac{1}{R}N_A & 0 & N_{A,x} & N_A & 0 \\ 0 & 0 & N_{A,y} & 0 & N_A \end{bmatrix}$$

and

$$B_A^N = AB_A^g = \begin{bmatrix} w_{0,x} & 0 \\ 0 & w_{0,y} \\ w_{0,y} & w_{0,x} \end{bmatrix} \begin{bmatrix} 0 & 0 & N_{A,x} & 0 & 0 \\ 0 & 0 & N_{A,y} & 0 & 0 \end{bmatrix}$$

where w_0 is the radial deflector of the panel.

FSDT on the Nonlinear Isogeometric Analysis of Shells

The FG-CNTRC Shell Formulation based on NURBS

$$K_T(q_m)\Delta^i q_m = {}^iF_{\text{ext},m} - {}^iF_{\text{int},m}$$

where at the ith iteration and mth load increment

- q_m is the displacement vector
- $\Delta^i q_m$ is the change in load vector
- ${}^iF_{\text{ext},m}$ is the load vector
- ${}^iF_{\text{int},m}$ is the internal force

FSDT on the Nonlinear Isogeometric Analysis of Shells

The FG-CNTRC Shell Formulation based on NURBS

$$K_T(q_m)\Delta^i q_m = {}^iF_{\text{ext},m} - {}^iF_{\text{int},m}$$

where at the ith iteration and mth load increment

- q_m is the displacement vector
- $\Delta^i q_m$ is the change in load vector
- ${}^iF_{\text{ext},m}$ is the load vector
- ${}^iF_{\text{int},m}$ is the internal force

FSDT on the Nonlinear Isogeometric Analysis of Shells

The FG-CNTRC Shell Formulation based on NURBS

$$K_T(q_m)\Delta^i q_m = {}^iF_{\text{ext},m} - {}^iF_{\text{int},m}$$

where at the ith iteration and mth load increment

- q_m is the displacement vector
- $\Delta^i q_m$ is the change in load vector
- ${}^iF_{\text{ext},m}$ is the load vector
- ${}^iF_{\text{int},m}$ is the internal force

FSDT on the Nonlinear Isogeometric Analysis of Shells

The FG-CNTRC Shell Formulation based on NURBS

$$K_T(q_m)\Delta^i q_m = {}^iF_{\text{ext},m} - {}^iF_{\text{int},m}$$

where at the ith iteration and mth load increment

- q_m is the displacement vector
- $\Delta^i q_m$ is the change in load vector
- ${}^iF_{\text{ext},m}$ is the load vector
- ${}^iF_{\text{int},m}$ is the internal force

FSDT on the Nonlinear Isogeometric Analysis of Shells

Using the novel OINS to solve nonlinear equation, for any load increments, an iterative process is conducted and will stop when the convergence criterion e is met as follows

Convergence Criterion e

$$e = \frac{\|{}^{i}\lambda_m F_0 - {}^{i}F_{\text{int},m}\|}{\|({}^{i}\lambda_m + \Delta^{i}\lambda_m)F_0\|} < 10^{-3}$$

where λ is denoted as the load factor.

FSDT on the Nonlinear Isogeometric Analysis of Shells

Using the novel OINS to solve nonlinear equation, for any load increments, an iterative process is conducted and will stop when the convergence criterion e is met as follows

Convergence Criterion e

$$e = \frac{\|{}^{i}\lambda_m F_0 - {}^{i}F_{\mathsf{int},m}\|}{\|({}^{i}\lambda_m + \Delta {}^{i}\lambda_m) F_0\|} < 10^{-3}$$

where λ is denoted as the load factor.

Convergence on Iterative Process

Note that the load vector at $i$the iteration and mth load increment

$$ {}^{i}F_{\mathsf{ext},m} = \left({}^{i}\lambda_m + \Delta^{i}\lambda_m\right) \int_{\Omega} f_0 \left\{0 \quad 0 \quad N_A \quad 0 \quad 0\right\}^T d\Omega = \left({}^{i}\lambda_m + \Delta^{i}\lambda_m\right) F_0 \qquad (1) $$

where F_0 denotes the referenced load vector while N_A stands for the NURBS (nonuniform rational B-spline) basic function.

Convergence on Iterative Process

The load factor λ and the displacement vector q of the iteration are updated as follows

$$\begin{aligned} {}^{i+1}\lambda_m &=^i \lambda_m + \Delta^i\lambda_m \\ {}^{i+1}q_m &= {}^iq_m + \Delta^iq_m \\ \Delta^iq_m &= \Delta^iq_{R,m} + \Delta^i\lambda_m q_{F,m} \end{aligned}$$

Moreover, $q_{F,m}$ and $\Delta^iq_{R,m}$ respectively denote the displacement created by the reference vector and the change caused by the residual load vectors and that

$$\Delta^iq_{R,m} = [K_T(q_m)]^{-1}\left({}^i\lambda_m F_0 - {}^iF_{\text{int},m}\right) \quad (2)$$

$$q_{F,m} = [K_T(q_m)]^{-1}F_0 \quad (3)$$

Recall that $K_T(q_m)\Delta^iq_m = {}^iF_{\text{ext},m} - {}^iF_{\text{int},m}$.

Convergence on Iterative Process

The load factor λ and the displacement vector q of the iteration are updated as follows

$$
\begin{aligned}
{}^{i+1}\lambda_m &=^{i} \lambda_m + \Delta^{i}\lambda_m \\
{}^{i+1}q_m &= {}^{i}q_m + \Delta^{i}q_m \\
\Delta^{i}q_m &= \Delta^{i}q_{R,m} + \Delta^{i}\lambda_m q_{F,m}
\end{aligned}
$$

Moreover, $q_{F,m}$ and $\Delta^{i}q_{R,m}$ respectively denote the displacement created by the reference vector and the change caused by the residual load vectors and that

$$
\Delta^{i}q_{R,m} = [K_T(q_m)]^{-1}\left({}^{i}\lambda_m F_0 - {}^{i}F_{\text{int},m}\right) \tag{2}
$$

$$
q_{F,m} = [K_T(q_m)]^{-1}F_0 \tag{3}
$$

Recall that $K_T(q_m)\Delta^{i}q_m = {}^{i}F_{\text{ext},m} - {}^{i}F_{\text{int},m}$.

Convergence on Iterative Process

The load factor λ and the displacement vector q of the iteration are updated as follows

$$\begin{aligned} {}^{i+1}\lambda_m &=^i \lambda_m + \Delta^i\lambda_m \\ {}^{i+1}q_m &= {}^i q_m + \Delta^i q_m \\ \Delta^i q_m &= \Delta^i q_{R,m} + \Delta^i \lambda_m q_{F,m} \end{aligned}$$

Moreover, $q_{F,m}$ and $\Delta^i q_{R,m}$ respectively denote the displacement created by the reference vector and the change caused by the residual load vectors and that

$$\Delta^i q_{R,m} = [K_T(q_m)]^{-1} \left({}^i\lambda_m F_0 - {}^i F_{\text{int},m}\right) \tag{2}$$

$$q_{F,m} = [K_T(q_m)]^{-1} F_0 \tag{3}$$

Recall that $K_T(q_m)\Delta^i q_m = {}^i F_{\text{ext},m} - {}^i F_{\text{int},m}$.

Convergence on Iterative Process

Rewriting equations (2), we have

$$K_T(q_m)\Delta^i q_{R,m} = \left({}^i\lambda_m F_0 - {}^i F_{\mathsf{int},m}\right) \tag{4}$$

then we can rewrite the convergence criterion e as follows

$$\begin{aligned} e &= \frac{\|{}^i\lambda_m F_0 - {}^i F_{\mathsf{int},m}\|}{\|({}^i\lambda_m + \Delta^i\lambda_m)F_0\|} \\ &= \frac{\|K_T(q_m)\Delta^i q_{R,m}\|}{\|{}^i F_{\mathsf{ext},m}\|} \quad [1,4] \end{aligned}$$

Convergence on Iterative Process

Rewriting equations (2), we have

$$K_T(q_m)\Delta^i q_{R,m} = \left({}^i\lambda_m F_0 - {}^i F_{\text{int},m}\right) \tag{4}$$

then we can rewrite the convergence criterion e as follows

$$\begin{aligned} e &= \frac{\|{}^i\lambda_m F_0 - {}^i F_{\text{int},m}\|}{\|({}^i\lambda_m + \Delta^i\lambda_m)F_0\|} \\ &= \frac{\|K_T(q_m)\Delta^i q_{R,m}\|}{\|{}^i F_{\text{ext},m}\|} \quad [1,4] \end{aligned}$$

Convergence on Iterative Process

Rewriting equations (2), we have

$$K_T(q_m)\Delta^i q_{R,m} = \left({}^i\lambda_m F_0 - {}^i F_{\mathsf{int},m}\right) \tag{4}$$

then we can rewrite the convergence criterion e as follows

$$\begin{aligned} e &= \frac{\|{}^i\lambda_m F_0 - {}^i F_{\mathsf{int},m}\|}{\|({}^i\lambda_m + \Delta^i\lambda_m)F_0\|} \\ &= \frac{\|K_T(q_m)\Delta^i q_{R,m}\|}{\|{}^i F_{\mathsf{ext},m}\|} \quad [1,4] \end{aligned}$$

Outline

Goals/Aims

To overcome completely this difficulty, the novel one-iteration nonlinear solver (OINS) using time series prediction and the modified Riks method (M-R) is proposed in this paper.

Goals/Aims

To overcome completely this difficulty, the novel one-iteration nonlinear solver (OINS) using time series prediction and the modified Riks method (M-R) is proposed in this paper.

Modifield Riks (M-R) Method

The **Modified Riks (M-R) Method** is a nonlinear problem solver that uses iterative process to come up to the converging solution.

Modifield Riks (M-R) Method

BeamerTemplateforBUCT/Figures/Modified Riks Method.png

One-iteration Nonlinear Solver (OINS)

The **One-iteration Nonlinear Solver** is the proposed solver of the study that is developed from the M-R method implemented with the Group Method of Data Handling (GMDH)

One-iteration Nonlinear Solver

```
BeamerTemplateforBUCT/Figures/OINS Method.png
```

OINS-GMDH Training Data

BeamerTemplateforBUCT/Figures/OINS 8th Iteration.png

Outline

Methodology

OINS is established upon the core idea as follows: (1) Firstly, we predict the load factor increment and the displacement vector increment and the convergent solution of the considering load step via the predictive networks which are trained by using the load factor and the displacement vector increments of the previous convergence steps and group method of data handling (GMDH);(2) Thanks to the predicted convergence solution of the load step is very close to or identical with the real one, the prediction phase used in any existing nonlinear solvers is eliminated completely in OINS. Next, the correction phase of the M-R is adopted and the OINS iteration is started at the predicted convergence point to reach the convergent solution. The training process and the applying process of GMDH are continuously conducted and repeated during the nonlinear analysis in order to predict the convergence point at the beginning of each load step.

Methodology

OINS is established upon the core idea as follows: (1) Firstly, we predict the load factor increment and the displacement vector increment and the convergent solution of the considering load step via the predictive networks which are trained by using the load factor and the displacement vector increments of the previous convergence steps and group method of data handling (GMDH);(2) Thanks to the predicted convergence solution of the load step is very close to or identical with the real one, the prediction phase used in any existing nonlinear solvers is eliminated completely in OINS. Next, the correction phase of the M-R is adopted and the OINS iteration is started at the predicted convergence point to reach the convergent solution. The training process and the applying process of GMDH are continuously conducted and repeated during the nonlinear analysis in order to predict the convergence point at the beginning of each load step.

Group Method of Data Handling

The Group Method of Data Handling (GMDH) is a self-organizing deep learning method which is commonly and widely used for time series prediction problems. During the training stages, the number of neurons of GMDH networks continuously changes to improve the performance of the network.

BeamerTemplateforBUCT/Figures/GMDH Training.png

Methodology

OINS is established upon the core idea as follows: (1) Firstly, we predict the load factor increment and the displacement vector increment and the convergent solution of the considering load step via the predictive networks which are trained by using the load factor and the displacement vector increments of the previous convergence steps and group method of data handling (GMDH);(2) Thanks to the predicted convergence solution of the load step is very close to or identical with the real one, the prediction phase used in any existing nonlinear solvers is eliminated completely in OINS. Next, the correction phase of the M-R is adopted and the OINS iteration is started at the predicted convergence point to reach the convergent solution. The training process and the applying process of GMDH are continuously conducted and repeated during the nonlinear analysis in order to predict the convergence point at the beginning of each load step.

Outline

Results and Conclusion

Through numerical investigations, we prove that OINS is powerful, highly accurate and only needs about one iteration per load step. Thus, OINS significantly saves number of iterations and a huge amount of computation compared with the conventional methods.

The Star Players

The study compared the three nonlinear problem solvers namely:

1. The M-R Method
2. The Data-driven Nonlinear Solver (DDNS) Method
3. The OINS Method

The study conducted an iterative comparison between these solvers showing their efficiency in solving nonlinear problem.

The DDNS Method

Data-driven Nonlinear Solver has been developed recently to reduce the number of iterations of nonlinear problems by having the same idea to that of OINS method: predict.

The difference between DDNS and OINS? The DDNS predicts the new starting point closer to the convergent solution while OINS predicts the converging solution which cuts a large iteration in the whole process.

BeamerTemplateforBUCT/Figures/DDNS vs OINS.png

The DDNS Method

Data-driven Nonlinear Solver has been developed recently to reduce the number of iterations of nonlinear problems by having the same idea to that of OINS method: predict.
The difference between DDNS and OINS? The DDNS predicts the new starting point closer to the convergent solution while OINS predicts the converging solution which cuts a large iteration in the whole process.

BeamerTemplateforBUCT/Figures/DDNS vs OINS.png

The Star Players

As a result, the study has shown that the OINS method was 77-80% faster than the M-R method and DDNS, based on literature, is 40-50% faster than M-R method.

`BeamerTemplateforBUCT/Figures/OINS vs DDNS vs MR.png`

M-R vs OINS, continued

The Modified Riks and OINS method were tested using the problem of cylindrical panel.

BeamerTemplateforBUCT/Figures/Cylindrical hinged panel under a p

M-R vs OINS, continued

Central deflection of a hinged panel with $h = 25.4mm, B = 508mm, L = B$

BeamerTemplateforBUCT/M-R vs OINS 1.png

BeamerTemplateforBUCT/M-R vs OINS 1 fig.png

M-R vs OINS, continued

Central deflection of a hinged panel with $h = 6.35mm, B = 508mm, L = B$

BeamerTemplateforBUCT/M-R vs OINS 2 .png

BeamerTemplateforBUCT/M-R vs OINS 2 fig.png

Outline

Impications

Especially, OINS not only can detect limit, inflection, and other special points but also can predict exactly various types of instabilities of structures

Research Problem

It was noted that the solvers were applied to nonlinear problems in solid mechanics using the First-order Sheer Deformation Theory analysis of shells by the strain vectors. Given that the OINS method is proved to be efficient and accurate, we will attempt to apply the method to solve nonlinear problems in fluid mechanics by finding a set of parameters and equations that can be used in finite element analysis using OINS. Moreover, if possible, we will try to implement other deep learning method for time series prediction that might eliminate the corrective phase of the OINS method.

BibTex

[?] Data-driven Nonlinear Solver
[?] NURBS-based postbuckling analysis.

References