

COMPARISON OF COMPUTER-AIDED INTEGRATION TECHNIQUES WHEN OBTAINING INITIAL DISPLACEMENTS MATRIX FOR GEOMETRICALLY NON-LINEAR LAMINATED FINITE ELEMENT

V. Kulinič MSc (Eng)

To cite this article: V. Kulinič MSc (Eng) (2001) COMPARISON OF COMPUTER-AIDED INTEGRATION TECHNIQUES WHEN OBTAINING INITIAL DISPLACEMENTS MATRIX FOR GEOMETRICALLY NON-LINEAR LAMINATED FINITE ELEMENT, *Statyba*, 7:5, 352-358, DOI: [10.1080/13921525.2001.10531753](https://doi.org/10.1080/13921525.2001.10531753)

To link to this article: <https://doi.org/10.1080/13921525.2001.10531753>



Published online: 30 Jul 2012.



Submit your article to this journal [↗](#)



Article views: 35

KOMPIUTERINIŲ INTEGRAVIMO TECHNOLOGIJŲ PALYGINIMAS SUDARANT GEOMETRIŠKAI NETIESINIO SLUOKSNIUOTOJO BAIGTINIO ELEMENTO PRADINIŲ POSLINKIŲ MATRICĄ

V. Kulinič

Vilniaus Gedimino technikos universitetas

1. Įvadas

Statyboje ir kitose technikos srityse pastaruosiu metu naudojama vis daugiau kompozitinių ortotropinių konstrukcinių elementų. Sprendžiant pakankamai plonų sluoksniuotųjų statybinės ir kitos paskirties plokščių skaičiavimo uždavinius, rezultatai dažnai būna netikslūs, jeigu laikomasi prielaidos, kad konstrukcijos elementų poslinkiai ir deformacijos yra labai maži.

Daugelyje žinomų komercinių ir mokslinių skaičiavimo paketų, tokių kaip ABAQUS, ALGOR, ANSYS, COSMOS/M, nerealizuotas geometrinis netiesiškumas sluoksniuotoms plokštėms [1–4]. Šia kryptimi vykdomi tyrimai ir yra sukurta daug programų, tačiau kompozitinėms (sluoksniuotosioms) konstrukcijoms modeliuoti dažniausiai kuriamos tokios metodikos, kurias galima sėkmingai taikyti tik tam tikrais konkrečiais atvejais [5].

Vienintelis bendrojo pobūdžio baigtinis elementas sluoksniuotosioms ortotropinėms lenkiamosioms plokštėms, kuriam nekeliami jokie formos, sluoksnių storių, medžiagų fizikinių bei mechaninių charakteristikų, ortotropiškumo ašių krypties, apkrovimo būdo ir kraštinių sąlygų reikalavimai, yra TRIPLT [6]. Tačiau iki šiol šiam pakankamai sudėtingam elementui, turinčiam 50 laisvės laipsnių, nebuvo realizuotas geometrinis netiesiškumas.

Atliekamų tyrimų tikslas – pagal elementą TRIPLT sukurti universalų geometriškai netiesinį sluoksniuotąjį elementą, kuris gali būti naudojamas įvairiausiomis (įskaitant ir gelžbetonines) plokštėms modeliuoti.

Realizuojant geometrinį netiesiškumą daugiausia dėmesio skiriama liestinei standumo matricai, kuri yra svarbiausia vykstant iteraciniams procesams. Kadangi kiekviename iteracijos žingsnyje su ja atliekami įvairūs

veiksmai, todėl galima teigti, kad skaičiavimo modelio efektyvumas labiausiai priklausys nuo liestinės standumo matricos sudarymo ir apdorojimo algoritmų efektyvumo. Šioms matricoms sudaryti taikoma kompiuterinės algebros technologija.

Straipsnyje parodyta, kaip žinomam tiesiniam elementui TRIPLT pritaikomas geometrinis netiesiškumas, apibūdinta liestinė standumo matrica geometriškai netiesiniam elementui ir pateiktos pradinių poslinkių matricos išraiškos.

Paprogramiai pradinių poslinkių matricai skaičiuoti išvesti taikant kompiuterinę algebrą. Straipsnyje aprašytos kompiuterinio integravimo procedūros ir lyginami integravimo metodai.

2. Baigtinis elementas TRIPLT

Sluoksniuotojo baigtinio elemento TRIPLT geometrinės matricos gaunamos naudojant normuotąsias L -koordinates [7, 8]. Poslinkiai ir posūkliai elemento viduje išreiškiami, naudojant mazginius poslinkius (posūkius), jų gradientus ir poslinkius (posūkius) elemento viduje [7].

Atsižvelgiant į tiesinę ir netiesinę dedamąsias, sluoksniuotojo baigtinio elemento TRIPLT deformacijas galima užrašyti taip:

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \boldsymbol{\varepsilon}_0^{pl} \\ \boldsymbol{\varepsilon}_0^b \\ \boldsymbol{\varepsilon}_0^s \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{\varepsilon}_L^{pl} \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} = \left(\begin{Bmatrix} [B^{pl}] \\ [B^b] \\ [B^s] \end{Bmatrix} + \begin{Bmatrix} [B_L^{pl}] \\ [0] \\ [0] \end{Bmatrix} \right) \boldsymbol{\psi} = [\mathcal{B}] \boldsymbol{\psi}, \quad (1)$$

$\boldsymbol{\varepsilon}_0^{pl}$ – tiesinė membraninių deformacijų dalis; $\boldsymbol{\varepsilon}_0^b$ – lenkimo deformacijos (kreiviai); $\boldsymbol{\varepsilon}_0^s$ – šlyties deformacijos; $\boldsymbol{\varepsilon}_L^{pl}$ – netiesinė membraninių deformacijų dalis; $\boldsymbol{\psi} = \{\boldsymbol{\psi}_u, \boldsymbol{\psi}_v, \boldsymbol{\psi}_w, \boldsymbol{\psi}_{\theta 1}, \boldsymbol{\psi}_{\theta 2}\}^T$ – visų mazginių poslinkių ir posūkių vektorius;

$$\begin{bmatrix} B^{pl} \\ B^b \\ B^s \end{bmatrix} = [B_L^e] - \text{geometrinės matricos tiesinė da-}$$

lis, kuri susideda iš trijų submatricių, atitinkančių membranines, lenkimo ir šlyties deformacijas:

$$\begin{bmatrix} \mathbf{H}_{,1}^T [F]^{-1} & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{H}_{,2}^T [F]^{-1} & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{H}_{,2}^T [F]^{-1} & \mathbf{H}_{,1}^T [F]^{-1} & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & -\mathbf{H}_{,1}^T [F]^{-1} & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & -\mathbf{H}_{,2}^T [F]^{-1} \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & -\mathbf{H}_{,2}^T [F]^{-1} & -\mathbf{H}_{,1}^T [F]^{-1} \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{H}_{,2}^T [F]^{-1} & \mathbf{0}^T & -\mathbf{H}^T [F]^{-1} \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{H}_{,1}^T [F]^{-1} & -\mathbf{H}^T [F]^{-1} & \mathbf{0}^T \end{bmatrix} \quad (2)$$

$\mathbf{H}^T = \{L_1^3, L_2^3, L_3^3, L_1^2 L_2, L_2^2 L_3, L_3^2 L_1, L_1 L_2^2, L_2 L_3^2, L_3 L_1^2, L_1, L_2, L_3\}$ – normuotųjų koordinačių

vektorius; $\mathbf{H}_{,1}^T \equiv \frac{\partial \mathbf{H}^T}{\partial \mathbf{x}}, \mathbf{H}_{,2}^T \equiv \frac{\partial \mathbf{H}^T}{\partial \mathbf{y}}$ – jo išvestinės;

$[F]$ – konfigūracijos matrica, kurios išraišką galima rasti literatūroje [7];

$$\begin{bmatrix} B_L^{pl} \\ [0] \\ [0] \end{bmatrix} = [B_L^e] - \text{geometrinės matricos netiesinė dalis.}$$

Remiantis geometriškai netiesinių baigtinių elementų teorija [7], netiesinė membraninių deformacijų dalis gali būti užrašyta taip:

$$\varepsilon_L^{pl} = \frac{1}{2} \begin{bmatrix} \mathbf{H}_{,1}^T [F]^{-1} \boldsymbol{\psi}_w & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{,2}^T [F]^{-1} \boldsymbol{\psi}_w \\ \mathbf{H}_{,2}^T [F]^{-1} \boldsymbol{\psi}_w & \mathbf{H}_{,1}^T [F]^{-1} \boldsymbol{\psi}_w \end{bmatrix} \times \begin{bmatrix} \mathbf{H}_{,1}^T [F]^{-1} \\ \mathbf{H}_{,2}^T [F]^{-1} \end{bmatrix} \boldsymbol{\psi}_w = [B_L^{pl}]^* \boldsymbol{\psi}_w. \quad (3)$$

Taip gaunama geometrinės matricos netiesinės dalies $[B_L^e]$ submatrica:

$$[B_L^{pl}]^* = \begin{bmatrix} \mathbf{H}_{,1}^T [F]^{-1} \boldsymbol{\psi}_w \mathbf{H}_{,1}^T [F]^{-1} \\ \mathbf{H}_{,2}^T [F]^{-1} \boldsymbol{\psi}_w \mathbf{H}_{,2}^T [F]^{-1} \\ \mathbf{H}_{,2}^T [F]^{-1} \boldsymbol{\psi}_w \mathbf{H}_{,1}^T [F]^{-1} + \mathbf{H}_{,1}^T [F]^{-1} \boldsymbol{\psi}_w \mathbf{H}_{,2}^T [F]^{-1} \end{bmatrix} \quad (4)$$

Visi kiti matricos $[B_L^e]$ elementai yra nuliniai vektoriai:

$$[B_L^e] = \begin{bmatrix} \begin{bmatrix} \mathbf{0}^T \\ \mathbf{0}^T \\ \mathbf{0}^T \end{bmatrix} & \begin{bmatrix} \mathbf{0}^T \\ \mathbf{0}^T \\ \mathbf{0}^T \end{bmatrix} & [B_L^{pl}]^* & \begin{bmatrix} \mathbf{0}^T \\ \mathbf{0}^T \\ \mathbf{0}^T \end{bmatrix} & \begin{bmatrix} \mathbf{0}^T \\ \mathbf{0}^T \\ \mathbf{0}^T \end{bmatrix} \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \quad (5)$$

Jėgos ir momentai, veikiantys sluoksniuotojoje plokštėje, skaičiuojami pagal žinomas išraiškas [9, 10], o sluoksniuotosios plokštės membraninių, lenkimo ir šlyties įtempimų atstojamųjų vektorius gali būti užrašytas taip:

$$\boldsymbol{\sigma} = [\mathcal{L}] \begin{Bmatrix} \varepsilon_{x,0} + \varepsilon_{x,L} \\ \varepsilon_{y,0} + \varepsilon_{y,L} \\ \varepsilon_{xy,0} + \varepsilon_{xy,L} \\ -\kappa_x \\ -\kappa_y \\ -2\kappa_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{Bmatrix} = [\mathcal{L}] (\boldsymbol{\varepsilon}_0 + \boldsymbol{\varepsilon}_L) = [\mathcal{L}] \boldsymbol{\varepsilon}, \quad (6)$$

$[\mathcal{L}]$ – bendroji medžiagos tamprumo matrica:

$$[\mathcal{L}] = \begin{bmatrix} [A] & [B] & [0] \\ [B] & [D] & [0] \\ [0] & [0] & [\bar{A}] \end{bmatrix}. \quad (7)$$

Submatricos $[A]$, $[B]$, $[D]$ ir $[\bar{A}]$ gaunamos iš jėgų ir momentų išraiškų, naudojant integravimo procedūras [9, 10].

3. Liestinė standumo matrica

Remiantis baigtinių elementų teorija [7], geometriškai netiesinės baigtinių elementų sistemos lygtis yra:

$$[K_T(\boldsymbol{\psi})] \boldsymbol{\psi} = \mathbf{P}, \quad (8)$$

$[K_T(\boldsymbol{\psi})]$ – liestinė (tangentinė) standumo matrica, priklausanti nuo mazginių poslinkių vektoriaus $\boldsymbol{\psi}$ ir susidedanti iš įprastinės tiesinės tampriosios standumo matricos $[K_0]$ ir dviejų pakopinių standumo matricių:

pradinių įtempimų matricos $[K_{\sigma}]$, kuri priklauso nuo einamosios įtempimų būsenos, ir pradinių poslinkių (arba didelių poslinkių) matricos $[K_L]$, kuri tiesiškai ir kvadratiškai priklauso nuo poslinkių.

Baigtinio elemento pradinių poslinkių matrica:

$$[K_L^e] = \int_{\Delta} [B_0^e]^T [\mathcal{G}] [B_L^e] d\Delta + \int_{\Delta} [B_L^e]^T [\mathcal{G}] [B_0^e] d\Delta + \int_{\Delta} [B_L^e]^T [\mathcal{G}] [B_L^e] d\Delta \quad (9)$$

arba išreiškiant bendrosios tamprumo matricos submatricomis (7):

$$[K_L^e] = \int_{\Delta} [B^{pl}]^T [A] [B_L^b] d\Delta + \int_{\Delta} [B^b]^T [B] [B_L^b] d\Delta + \int_{\Delta} [B_L^b]^T [A] [B^{pl}] d\Delta + \int_{\Delta} [B_L^b]^T [B] [B^b] d\Delta + \int_{\Delta} [B_L^b]^T [A] [B_L^b] d\Delta \quad (10)$$

4. Pradinių poslinkių matrica

Pradinių poslinkių matricai gauti iš (10) naudojamos (10×10) dydžio bazinės matricos [7]:

$$[T_{i,jk}] = [F]^T \int_{\Delta} \mathbf{H}_{,i} \mathbf{H}_{,j}^T [F]^T \Psi_w \mathbf{H}_{,k}^T d\Delta [F]^T \quad \text{ir} \quad (11)$$

$$[T_{ij,kl}] = [F]^T \int_{\Delta} \mathbf{H}_{,i} \Psi_w^T [F]^T \mathbf{H}_{,j} \mathbf{H}_{,k}^T [F]^T \Psi_w \mathbf{H}_{,l}^T d\Delta [F]^T \quad (12)$$

($i, j, k, l=1, 2$).

Visa pradinių poslinkių matrica (50×50) susideda iš devynių (10×10) dydžio submatricų:

$$[K_L] = \begin{bmatrix} [0] & [0] & [K_{L13}] & [0] & [0] \\ & [0] & [K_{L23}] & [0] & [0] \\ & & [K_{L33}] & [K_{L34}] & [K_{L35}] \\ & & & [0] & [0] \\ \text{symm} & & & & [0] \end{bmatrix}, \quad (13)$$

čia:

$$[K_{L13}] = A_{11}[T_{1,11}] + A_{12}[T_{1,22}] + A_{26}[T_{2,22}] + A_{66}([T_{2,21}] + [T_{2,12}]) + A_{16}([T_{2,11}] + [T_{1,21}] + [T_{1,12}])$$

$$[K_{L23}] = A_{12}[T_{2,11}] + A_{16}[T_{1,11}] + A_{22}[T_{2,22}] + A_{66}([T_{1,21}] + [T_{1,12}]) + A_{26}([T_{1,22}] + [T_{2,21}] + [T_{2,12}])$$

$$[K_{L34}] = -B_{11}[T_{1,11}]^T - B_{12}[T_{1,22}]^T - B_{26}[T_{2,22}]^T - B_{66}([T_{2,21}]^T + [T_{2,12}]^T) - B_{16}([T_{2,11}]^T + [T_{1,21}]^T + [T_{1,12}]^T)$$

$$[K_{L35}] = -B_{12}[T_{2,11}]^T - B_{16}[T_{1,11}]^T - B_{22}[T_{2,22}]^T - B_{66}([T_{1,21}]^T + [T_{1,12}]^T) - B_{26}([T_{1,22}]^T + [T_{2,21}]^T + [T_{2,12}]^T)$$

$$[K_{L33}] = A_{11}[T_{1,11}] + A_{22}[T_{2,22}] + A_{12}([T_{1,1,22}]^T + [T_{1,1,22}]) + A_{16}([T_{1,1,21}]^T + [T_{1,1,12}]^T + [T_{1,1,21}] + [T_{1,1,12}]) + A_{26}([T_{1,2,22}] + [T_{2,1,22}] + [T_{1,2,22}]^T + [T_{2,1,22}]^T) + A_{66}([T_{1,2,21}] + [T_{1,2,12}]^T + [T_{1,2,12}] + [T_{2,1,12}]) \quad (14)$$

A_{ij} ir B_{ij} yra skaliariniai bendrosios tamprumo matricos (7) komponentai, o kokybiškas bazinių matricų išraiškas galima gauti tik taikant kompiuterinę algebrą [11], naudojant analizinį arba skaitinį integravimą.

5. Pradinių poslinkių matricos sudarymas, taikant kompiuterinę algebrą

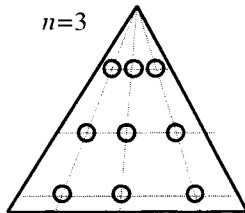
Bazinės matricos $[T_{i,jk}]$ ($i, j, k=1, 2$) ir $[T_{ij,kl}]$ ($i, j, k, l=1, 2$) gaunamos, suintegravus funkcijas, į kurias įeina atitinkamai šeštojo ir aštuntojo laipsnio L -koordinacių polinomiali, naudojant analizinį integravimą pagal formulę [7]:

$$\int_{\Delta} L_1 L_2 L_3 d\Delta = 2\Delta \frac{a!b!c!}{(a+b+c+2)!} \quad (15)$$

ir du skaitinio integravimo būdus, kurie toliau bus vadinami Gauso ir Radau metodu (naudojant Gauso ir

Radau svorio koeficientus [12]) ir Hammerio metodu (naudojant Hammerio svorio koeficientus [13]).

Šeštojo laipsnio polinomui skaitiškai suintegruoti reikia ne mažiau kaip septynių integravimo taškų, aštuntojo laipsnio polinomui – ne mažiau kaip devynių taškų [12]. Deja, taikant *Gauso ir Radau metodą* integruojama arba keturiuose, arba devyniuose taškuose, taigi abiem atvejais tenka imti devynis taškus (1 pav.).



1 pav. Taškai, skirti integruoti trikampio plote Gauso-Radau metodu

Fig 1. Integrating points in a triangular region by Gauss-Rada

Skaitinis integralas apskaičiuojamas pagal formulę [12]:

$$I = \sum_{i=1}^n \sum_{j=1}^n W_j f(L_1, L_2, L_3), \quad (16)$$

čia $f(L_1, L_2, L_3)$ – integruojama funkcija; mūsų atveju tai:

$$f(L_1, L_2, L_3) = \mathbf{H}_i \mathbf{H}_j^T [F]^T \Psi_w \mathbf{H}_k^T - \quad (17)$$

šeštojo laipsnio polinomų matrica, arba –

$$f(L_1, L_2, L_3) = \mathbf{H}_i \Psi_w^T [F]^T \mathbf{H}_j \mathbf{H}_k^T [F]^T \Psi_w \mathbf{H}_l^T - \quad (18)$$

aštuntojo laipsnio polinomų matrica.

W – svorio koeficientai:

$$W = AS(i)H(j)(1 - L_1). \quad (19)$$

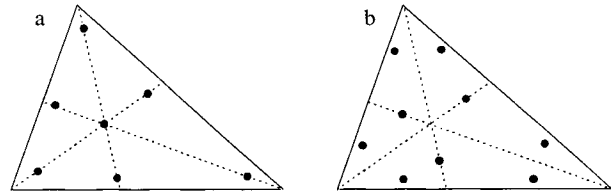
Integravimo taškų koordinatės:

$$\begin{aligned} L_1 &= AI(i), \\ L_2 &= AJ(j)(1 - L_1), \\ L_3 &= 1 - L_1 - L_2, \\ i, j &= 1, 2, 3. \end{aligned} \quad (20)$$

Funkcijų $AI(i)$, $AJ(j)$, $AS(i)$, $H(j)$ reikšmės imamos iš lentelių [12].

Integruojant *Hammerio metodu* šeštojo laipsnio polinomui imami septyni taškai (2 pav., a), o aštuntojo

laipsnio polinomui – devyni (2 pav., b).



2 pav. Taškai, skirti integruoti trikampio plote Hammerio metodu: a) šeštojo laipsnio polinomui, b) aštuntojo laipsnio polinomui

Fig 2. Integrating points in a triangular region by Hammer: a) for sixth degree polynomial, b) for eighth degree polynomial

Skaitinio integravimo formulė [13]:

$$I = \sum_{i=1}^n W_i f_i(L_1, L_2, L_3), \quad (21)$$

$n = 7$ – šeštojo laipsnio polinomui; $n = 9$ – aštuntojo laipsnio polinomui.

Hammerio svorio koeficientai W_i ir taškų koordinatės imami iš lentelių [13].

Kaip matyti iš lentelės, kurioje pateiktas metodų efektyvumo palyginimas, tiek integruojant analiziškai, tiek ir skaitiškai, reikia atlikti labai daug aritmetinių veiksmų, nors skaitinio integravimo atveju jų reikia žymiai mažiau.

Verta pažymėti, kad pradinių poslinkių submatricas išreikšti, naudojant bazines matricas, reikia tik analiziniam integravimui, nes kiekvienas polinomo narys turi būti integruojamas atskirai. Tačiau, integruojant skaitiškai, sumuojamos pačios integruojamos funkcijos, taigi naudoti šiam tikslui išplėstas submatricas $[K_{Lij}]$ (14) būtų labai neracionalu. Šiuo atveju reikia siekti kuo didesnio kompaktiškumo jau normuotųjų koordinatžių ir mazginių poslinkių vektorių lygiu.

Pavyzdžiui, galima išskirti skaliarines dalis:

$$\Psi_w^T \mathbf{F}^{-T} \mathbf{H}_i = \mathbf{H}_i^T \mathbf{F}^{-1} \Psi_w = S_i \quad (22)$$

ir apskaičiuoti jas atskirai kiekvienam integravimo taškui, iš integravimo lentelių įrašius atitinkamas integravimui paruoštas \mathbf{H}_1 ir \mathbf{H}_2 reikšmes ir koordinates, taip formuojant vektorius, kurių kiekvienas elementas yra skaliaro S_i reikšmė vienam iteracijos žingsniui.

Savo ruožtu atskirais paprogramiais kuriamos ir (10×9) dydžio matricos \mathbf{H}_1 ir \mathbf{H}_2 kurių stulpeliai

Integravimo metodų efektyvumo palyginimas, atsižvelgiant į gautų Fortrano paprogramių slankiojo kabelio operacijų skaičių
 Comparison of effectiveness of the integrating methods based on amount of floating point operations in FORTRAN subroutines

Matrica	Būdai				
	Analizinis	Gausso ir Radau		Hammerio	
	operacijų	taškų	operacijų	taškų	operacijų
Naudojant bazines matricas $[T_{ijk}]$ ir $[T_{i,kl}]$					
$[T_{ijk}]$	$139 \cdot 10^3$	9	$102 \cdot 10^3$	7	$80 \cdot 10^3$
$[T_{i,kl}]$	$1752 \cdot 10^3$	9	$197 \cdot 10^3$	9	$197 \cdot 10^3$
$[K_L]$	$18,6 \cdot 10^6$		$2,8 \cdot 10^6$		$2,6 \cdot 10^6$
Naudojant tarpinius vektorius ir matricas $[\mathbf{H}_i]$, $\{S_i\}$, $[\mathbf{H}_i S_i]$, $[M_{HS_{Ai}}]$					
$[\mathbf{H}_i]$				7	193
				9	237
$\{S_i\}$				7	746
				9	950
$[\mathbf{H}_i S_i]$				7	75
				9	93
$[M_{HS_{Ai}}]$				9	634
$[K_{Li3}]$ $i=1, 2, 4, 5$				7	60 998 *
				9	78 378 *
$[K_{L33}]$				9	11 859
$[K_L]$				7 ir 9	72 857
				9	87 011

* visoms keturioms matricoms skaičiuoti

* to have all four matrices calculated

yra normuotųjų koordinačių vektoriai \mathbf{H}_1 ir \mathbf{H}_2 , įrašius reikšmes iš integravimo lentelių. Po tam tikrų pertvarkymų matrica $[K_{L33}]$ yra tokia:

$$[K_{L33}] = \int_{\Delta} [B_L^b]^T [A] [B_L^b] d\Delta = [F]^{-T} \times$$

$$\int_{\Delta} \left(\begin{array}{l} (\mathbf{H}_1 S_1 A_{11} + \mathbf{H}_2 S_2 A_{12} + \\ + (\mathbf{H}_1 S_2 + \mathbf{H}_2 S_1) A_{16}) S_1 \mathbf{H}_1^T + \\ + (\mathbf{H}_1 S_1 A_{12} + \mathbf{H}_2 S_2 A_{22} + \\ + (\mathbf{H}_1 S_2 + \mathbf{H}_2 S_1) A_{26}) S_2 \mathbf{H}_2^T + \\ + (\mathbf{H}_1 S_1 A_{16} + \mathbf{H}_2 S_2 A_{26} + \\ + (\mathbf{H}_1 S_2 + \mathbf{H}_2 S_1) A_{66}) (S_1 \mathbf{H}_2^T + \\ + S_2 \mathbf{H}_1^T) \end{array} \right) d\Delta [F]^{-1}. \quad (23)$$

Kadangi sandaugos $\mathbf{H}_i S_j$, ($i, j = 1, 2$) įeina į (23) išraišką po penkis kartus, jas taip pat verta suskaičiuoti iš anksto. Tai irgi bus (10×9) dydžio matricos. Submatricos $[K_{L33}]$ skaičiavimą galima dar labiau supaprastinti, kadangi ji yra simetriška.

Panašūs pertvarkymai atliekami su kitomis $[K_{Lij}]$ submatricomis. Kadangi matrica $[K_L]$ yra simetriška, be $[K_{L33}]$, belieka suskaičiuoti keturias submatricas. Submatricos $[K_{L13}]$ išraiška, naudojant skaitinį integravimą:

$$[K_{L13}] = \int_{\Delta} [B^{pl}]^T [A] [B_L^b] d\Delta = [F]^{-T} \times$$

$$\int_{\Delta} \left(\begin{array}{l} (\mathbf{H}_1 S_1)_i^T (A_{11} (\mathbf{H}_1^T)_i + A_{16} (\mathbf{H}_2^T)_i) + \\ + (\mathbf{H}_1 S_2)_i^T (A_{12} (\mathbf{H}_2^T)_i + A_{16} (\mathbf{H}_1^T)_i) + \\ + (\mathbf{H}_2 S_1)_i^T (A_{66} (\mathbf{H}_2^T)_i + A_{16} (\mathbf{H}_2^T)_i) + \\ + (\mathbf{H}_2 S_2)_i^T (A_{26} (\mathbf{H}_2^T)_i + A_{66} (\mathbf{H}_1^T)_i) \end{array} \right) w_i [F]^{-1}, \quad (24)$$

w_i – Hammerio svorio koeficientai; $n = 7$ arba $n = 9$ priklausomai nuo pasirinkto integravimo tikslumo. Iš pirmo žvilgsnio gali atrodyti, jog būtų racionaliau submatricoms $[K_{L13}]$, $[K_{L23}]$, $[K_{L34}]$ ir $[K_{L35}]$ naudoti tas pačias $[\mathbf{H}_i]$, $\{S_i\}$, $[\mathbf{H}_i S_j]$ ($i, j = 1, 2$) išraiškas, ku-

rios buvo naudojamos submatricai $[K_{L33}]$ gauti, t. y. naudoti didesnį, negu reikia, integravimo tikslumą ($n = 9$), kad nereikėtų perskaičiuoti minėtų tarpinių vektorių ir matricų, kai $n = 7$.

Tačiau, naudojant MATHEMATICA, išvedus reikalingus paprogramius, paaiškėjo (žr. lentelę), kad, kai integravimas yra pakankamai tikslus (t. y. atskirai skaičiuojami tarpiniai vektoriai ir matricos, kai $n = 7$ ir kai $n = 9$), bendras operacijų skaičius matricai $[K_L]$ suskaičiuoti yra mažesnis.

6. Išvados

Taikant kompiuterinės algebros technologiją buvo integruoti šeštojo ir aštuntojo laipsnio polinomi ir gautos pradinių poslinkių matricos geometriškai netiesiniam sluoksniuotajam baigtiniam elementui TRIPLT.

Išanalizavus integravimo rezultatus, galima teigti kad:

1. Taikant skaitinį integravimą slankiojo kabelio operacijų skaičius yra pakankamai mažas. Tai leidžia realizuoti programą plačiai prieinamomis kompiuterinėmis priemonėmis, taigi šiuo atžvilgiu nėra ypatingų kliūčių geometrinį netiesiškumą efektyviai taikyti TRIPLT tipo baigtiniam elementui.

2. Kompaktiškiausi paprogramiai gaunami polinoms integruojant skaitiškai, su Hammerio koeficientais, naudojant tarpinius vektorius ir matricas.

3. Šeštojo laipsnio polinomui integruoti nebūtina taikyti Gausso ir Radau metodą: integravimo tikslumui nereikia šiam elementui devynių taškų, bet šiuo atveju devyni taškai – tai minimumas, kuris būtinas taikant šį integravimo metodą.

4. Aštuntojo laipsnio polinomo integravimo išraiškos turi beveik vienodą operacijų skaičių tiek taikant Gausso ir Radau metodą, tiek Hammerio. Kadangi taikant Hammerio metodą integravimo taškai trikampio plote išdėstomi simetriškai (1 pav. ir 2 pav. b), už Gausso ir Radau metodą jis turėtų būti efektyvesnis ir šiuo atveju.

5. Įvertinus tai, kad analizinio integravimo rezultatas yra tikslus (kiek leidžia kompiuterinis skaičiavimas), nebūtina visiškai jo atsisakyti. Jis gali būti taikomas, konstrukcijas skaičiuojant su galingais kompiuteriais, kurie gali atlikti kelis milijonus operacijų per sekundę.

Literatūra

1. ABAQUS/Explicit Product Description Version 5.8. URL: http://www.hks.com/products/p_abexplicit.html, 1999.
2. ALGOR. Physics-Based Mechanical Event Simulation and Nonlinear Stress Analysis. URL: <http://www.algor.com/homepag3.htm>, 1999.
3. ANSYS 5.5 Update: Quick Reference Guide. ANSYS/Structural. URL: <http://www.ansys.com/Products/Ansys55/whatisnew/str.html>, 1999.
4. COSMOS/M Version 1.75 for Windows NT & 95. Electronic documentation, 1998.
5. Modelling laminates. Fracture Research Group. University of Bristol. URL: <http://www.fen.bris.ac.uk/mech/research/material/projects/laminate.htm>, 1999.
6. H. V. Lakshminarayana, S. Sridhara Murthy. A Shear-flexible Triangular Finite Element Model for Laminated Composite Plates // Int. J. Numer. Meth. In Engng. 20, 1984, p. 591–623.
7. R. Belevičius. Computer Algebra in Finite Element Method. Vilnius: Technika, 1994. 184 p.
8. V. Kulinič. Non-Linear Analysis of Reinforced Concrete Beams Using Laminated Element. M.Sc. thesis. Vilnius, 1995. 70 p.
9. R. M. Jones. Mechanics of Composite Materials. McGraw-Hill, 1975. 383 p.
10. J. R. Vinson, R. L. Sierakowski. The Behavior of Structures Composed of Composite Materials. Martinus Nijhoff, 1986. 253 p.
11. В. П. Дьяконов. Системы символьной математики Mathematica 2 и Mathematica 3. Москва: СК ПРЕСС, 1998. 328 с.
12. O. C. Zienkiewicz, R. L. Taylor. The Finite Element Method. McGraw-Hill, 1971. 538 p.
13. G. R. Cowper. Gaussian Quadrature Formulas for Triangles // Int. J. Numer. Meth. in Engng., 1972, p. 1862–1868.

Įteikta 2001 10 18

COMPARISON OF COMPUTER-AIDED INTEGRATION TECHNIQUES WHEN OBTAINING INITIAL DISPLACEMENTS MATRIX FOR GEOMETRICALLY NON-LINEAR LAMINATED FINITE ELEMENT

V. Kulinič

Summary

Geometrical non-linearity of the laminated element has not been realized so far in the widely known commercial finite element method packages such as ABAQUS, ALGOR, ANSYS, COSMOS although researches in that field are actively carried out. On the other hand, there is a lot of problems where large displacements and deformations must be dealt with to obtain a precise decision.

A wide range of composite orthotropic materials is used in constructions and other fields of technology. Various numerical methods were implemented to handle laminated plates and shells, however most of them are intended for application only with particular types of the structures.

The author's aim is to develop a geometrically non-linear finite element that could be effectively used for analysis of various laminated slabs regardless of their shape, thickness of laminae, properties of materials, direction of orthotropy axes, way of loading and boundary conditions.

Obtaining and handling the element's initial displacement matrix used in the iteration process is a highly complicated issue requiring significant amount of computer resources to be involved. One of the most important aims of the research is to develop an element which could be used not only in an expensive multiprocessor mainframes, but also in an usual personal computer.

For the structure, a sophisticated finite element TRIPLT having 50 degrees of freedom is used. The geometrical matrix for this element is obtained involving L-coordinates' array while displacements and rotations in the middle of the element are expressed through the nodal displacements (rotations), their derivatives, and displacements (rotations) in the central point. Linear and non-linear components for the geometrical matrix are shown in Eqs 2 and 5.

The behaviour of a geometrical non-linear finite elements structure is described by Eq 8. The tangent stiffness matrix consists of the conventional linear elastic stiffness matrix, initial stress matrix and initial displacements matrix which is obtained by Eq 10, using both analytical and/or numerical integrating.

The analytical integrating involves expanding of the appropriate expressions into basic matrices (Eqs 11, 12) and using formula 15. The initial displacement matrix in term of constitutive matrix's elements and the basic matrices is shown in Eqs 13 and 14.

Numerical integrating is conducted by two methods: those using Hammer and Gauss-Radau weight coefficients. Numerical approach is applied both to the basic matrices and factorised expressions of submatrices involving intermediate arrays and matrices (Eqs 23, 24).

Two ways of obtaining the intermediate arrays and matrices are discussed.

Because of high complexity of the procedures involved the computer algebra system *Mathematica* was used for the integrating and recording FORTRAN codes.

Comparison of the effectiveness of all the procedures is presented in a table.

The investigation results show that the initial displacement matrix obtained by means of numerical integration involves a small amount of arithmetic operations to be handled with a usual personal computer.

.....
Valentinas KULINIČ. MSc (Eng). Vilnius Gediminas Technical University (VGTU), Saulėtekio al. 11, LT-2040 Vilnius, Lithuania. E-mail: valentinas@tilde.lt

A graduated of the VGTU (1994, civil engineer). MSc (1995). Project manager (Tilde IT Ltd.). Research interests: non-linear structural analysis, object-oriented and component-based finite element programming, mechanics of concrete.