Computer Applications in Electrical Engineering
Vol. 12 *2014*

Pre-assembly for FEM 2D non-curvilinear quadrilateral Lagrangian elements

Marcin Sowa
Silesian University of Technology
44-100 Gliwice, ul. Bolesława Krzywoustego 2

Dorota Typańska
Poznań University of Technology
60-965 Poznań, ul. Piotrowo 3a, e-mail: Dorota.Typanska@put.poznan.pl

The paper concerns an idea of preparing auxiliary equations and data for the finite element method assembly. This is made for a specific element type and element order and is referred to as the pre-assembly. Most importantly the prepared auxiliary equations require that only the element's placement (coordinates) needs to be known, while the remaining coefficients that are required are stored in data sheets. The pre-assembly allows programmers who implement the FEM to significantly reduce the effort put into the assembly algorithm through its replacement by a few simple equations and the application of the prepared data. These data sheets can be prepared and can be utilized by FEM programmers. The construction of these data sheets for non-curvilinear quadrilateral Lagrangian elements (of any selected order) is explained in this paper.

KEYWORDS: pre-assembly, finite element method, higher order elements, quadrilateral Lagrangian elements

1. Introduction

Generally an analysis with the use of the FEM (finite element method) can be divided into:

- definition of the problem – its geometry, boundary conditions, stimuli and environment parameters,
- meshing – which follows the choice of the geometrical elements e.g. as triangles or quadrilaterals,
- assembly – where the equations of the problem are formulated,
- problem solution – i.e. the actual computations of the field (the most significant computational weight is usually related to this part),
- post-processing – where the solution had been obtained and the results are interpreted.

In an electromagnetic field problem the system of equations is built basing on the weak formulation (which can be usually derived in general for a class of problems) and the mesh (along with the elements that it comprises of). In a traditional approach, when the element types have been selected, the local stiffness matrices are built for each element (note that for nonlinear analyses the dependencies are obtained instead). These are then added up to formulate the system of equations describing the problem. The most effort is put nowadays into reducing the computation time related to the solution of the formulated system of equations – even more so when the problem is time-dependent and the entire system of equations needs to be solved for each selected time instance.

In a time dependent problem the assembly part does not significantly contribute to the computation time, which is why there hasn't been much effort in optimizing it. However, if one deals with an object with moving parts e.g. induction motor [1] or for some other computational reasons (like when emphasizing adaptive mesh algorithms [2]) the meshing (and hence also the assembly) needs to be performed in every time-step of the analysis then it becomes worthwhile to optimize the assembly part.

This paper deals with a methodology where auxiliary formulae and data are prepared, which could be applied between the meshing and assembly, in order to simplify the assembly performed for each element. This preparation of formulae and data will be further on referred to as the pre-assembly.

The motivation of the pre-assembly is not only related to the time consumed for the assembly part but also it allows to reduce the effort put into the implementation of the FEM assembly. So far, in their previous paper [3], the authors have been successful in displaying the pre-assembly for non-curvilinear triangular Lagrange elements or arbitrary order, while this paper presents the case for non-curvilinear quadrilateral Lagrangian elements. The notation is as follows:

- m_g is the number of nodes representing the geometrical placement of the element (for non-curvilinear elements $m_g = 4$),
- m is the number of nodes for the interpolation of the component (this will be made by the shape functions described by N),
- $n = \sqrt{m}$ is the number of nodes on the element edge,
- n_o is the order of the element (which in case of the quadrilateral Lagrangian element equals $n - 1$).

For triangular elements the pre-assembly is based on exact formulae while for quadrilateral elements the pre-assembly will be based on approximate formulae.

2. Domain-level auxiliary equations

This section explains how an appropriate form of the problem's weak formulation is obtained before the most important part of the pre-assembly is performed i.e. the derivations for a selected element type.

Assuming a linear medium, a 2D transient magnetic field problem is defined by the equation for the magnetic vector potential component A:

$$\frac{1}{\mu}\vec{\nabla}^2 A - \gamma\frac{\partial A}{\partial t} = -J_{\mathrm{ext}}, \tag{1}$$

from which the weak formulation can be obtained [4, 5]:

$$\oint_{\partial\Omega}\frac{1}{\mu}\frac{\partial A}{\partial n}w\,\mathrm{d}\Gamma - \int_{\Omega}\frac{1}{\mu}\left(\frac{\partial A}{\partial x}\frac{\partial w}{\partial x}+\frac{\partial A}{\partial y}\frac{\partial w}{\partial y}\right)\mathrm{d}x\mathrm{d}y - \int_{\Omega}\gamma\frac{\partial A}{\partial t}w\,\mathrm{d}x\mathrm{d}y + \int_{\Omega}J_{\mathrm{ext}}w\,\mathrm{d}x\mathrm{d}y = 0, \tag{2}$$

where Ω is the considered domain, w is the so-called test function, J_{ext} is an externally enforced current density. $\frac{\partial A}{\partial n}$ is the derivative across the direction normal to the boundary $\partial\Omega$ (that covers the domain Ω) and Γ describes the curve along the domain boundary. For convenience, the terms in (2) are presented as:

$$\Lambda_{\Omega} = -\int_{\Omega}\frac{1}{\mu}\left(\frac{\partial A}{\partial x}\frac{\partial w}{\partial x}+\frac{\partial A}{\partial y}\frac{\partial w}{\partial y}\right)\mathrm{d}x\mathrm{d}y, \tag{3}$$

$$K_{\Omega} = -\int_{\Omega}\gamma\frac{\partial A}{\partial t}w\,\mathrm{d}x\mathrm{d}y + \int_{\Omega}J_{\mathrm{ext}}w\,\mathrm{d}x\mathrm{d}y, \tag{4}$$

$$d_{\Omega} = \oint_{\partial\Omega}\frac{1}{\mu}\frac{\partial A}{\partial n}w\,\mathrm{d}\Gamma. \tag{5}$$

When the integration of the weak formulation is divided into the sub-domains represented by each element, a transformation is performed from the global coordinates (x, y) to local ones denoted by (ξ, η). This can be done with the application of the Jacobian matrix of transformation:

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}. \tag{6}$$

In terms of (2) the most important relationships are:

$$\begin{bmatrix} \frac{\partial A}{\partial \xi} \\ \frac{\partial A}{\partial \eta} \end{bmatrix} = \boldsymbol{J}\begin{bmatrix} \frac{\partial A}{\partial x} \\ \frac{\partial A}{\partial y} \end{bmatrix}, \tag{7}$$

and:

$$\mathrm{d}x\mathrm{d}y = \det\boldsymbol{J}\,\mathrm{d}\xi\,\mathrm{d}\eta. \tag{8}$$

The above equations allow to write the terms (3) and (4) respectively as:

$$\Lambda_{\Omega} = -\int_{\Omega} \frac{1}{\mu} \left(\begin{array}{l} \left(\frac{\partial y}{\partial \eta} \frac{\partial A}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial A}{\partial \eta} \right) \left(\frac{\partial y}{\partial \eta} \frac{\partial w}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial w}{\partial \eta} \right) + \\ + \left(-\frac{\partial x}{\partial \eta} \frac{\partial A}{\partial \xi} + \frac{\partial x}{\partial \xi} \frac{\partial A}{\partial \eta} \right) \left(-\frac{\partial x}{\partial \eta} \frac{\partial w}{\partial \xi} + \frac{\partial x}{\partial \xi} \frac{\partial w}{\partial \eta} \right) \end{array} \right) \frac{1}{\det \boldsymbol{J}} \mathrm{d}\xi \mathrm{d}\eta, \quad (9)$$

and:

$$K_{\Omega} = -\int_{\Omega} \gamma \frac{\partial A}{\partial t} w \det \boldsymbol{J} \, \mathrm{d}\xi \mathrm{d}\eta + \int_{\Omega'} J_{\mathrm{ext}} w \det \boldsymbol{J} \, \mathrm{d}\xi \mathrm{d}\eta. \quad (10)$$

3. Element-level auxiliary equations

When the domain is divided into elements, one can formulate additional auxiliary expressions concerning the element's b-th DOF (degree of freedom) and a-th test function:

$$\Lambda_{a,b} = -\int_{\Omega'} \left(\begin{array}{l} \left(\frac{\partial y}{\partial \eta} \frac{\partial N_a}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial N_a}{\partial \eta} \right) \left(\frac{\partial y}{\partial \eta} \frac{\partial N_b}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial N_b}{\partial \eta} \right) + \\ + \left(-\frac{\partial x}{\partial \eta} \frac{\partial N_a}{\partial \xi} + \frac{\partial x}{\partial \xi} \frac{\partial N_a}{\partial \eta} \right) \left(-\frac{\partial x}{\partial \eta} \frac{\partial N_b}{\partial \xi} + \frac{\partial x}{\partial \xi} \frac{\partial N_b}{\partial \eta} \right) \end{array} \right) \frac{1}{\det \boldsymbol{J}} \mathrm{d}\xi \mathrm{d}\eta, \quad (11)$$

and:

$$K_{a,b} = -\int_{\Omega'} N_a N_b \det \boldsymbol{J} \, \mathrm{d}\xi \mathrm{d}\eta, \quad (12)$$

where N_i denotes the respective basis function (which equals 1 at $({}_N x_i, {}_N y_i)$ and 0 at all the other nodes) and Ω' is the base element area. The above equations are actually useful for any 2D element type where the environment parameters γ and μ are constant at least throughout the element. For any 2D element one can present a scalar function like the magnetic vector potential component A as:

$$A = \sum_{i=1}^{m} A_i N_i, \quad (13)$$

where m is the number of element nodes and A_i are the values of the scalar function at the node's coordinates $({}_N \xi_i, {}_N \eta_i)$. The global coordinates can be presented as:

$$x = \sum_{i=1}^{m_{\mathrm{g}}} x_i G_i, \quad y = \sum_{i=1}^{m_{\mathrm{g}}} y_i G_i, \quad (14)$$

where (x_i, y_i) are the element's subsequent essential vertices (required for the presentation of the element's placement) and G represent the geometrical basis functions. For a non-curvilinear quadrilateral element (Figure 1) $m_{\mathrm{g}} = 4$ and:

$$G_1 = \frac{1}{4}(1 - \xi)(1 + \eta), \quad (15)$$

$$G_2 = \frac{1}{4}(1+\xi)(1+\eta), \tag{16}$$

$$G_3 = \frac{1}{4}(1+\xi)(1-\eta), \tag{17}$$

$$G_4 = \frac{1}{4}(1-\xi)(1-\eta), \tag{18}$$

which are actually the same as the component basis functions N for $m = 4$ (the G functions can be generally built in the same way as the N functions).

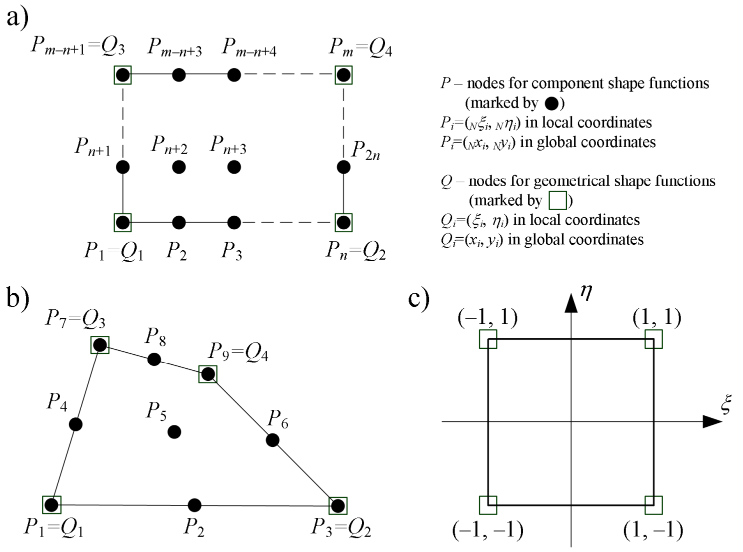

a)
$P_{m-n+1}=Q_3$
P_{m-n+3}
P_{m-n+4}
$P_m=Q_4$
P_{n+1}
P_{n+2}
P_{n+3}
P_{2n}
$P_1=Q_1$
P_2
P_3
$P_n=Q_2$
P – nodes for component shape functions (marked by ●)
$P_i=({}_N\xi_i, {}_N\eta_i)$ in local coordinates
$P_i=({}_Nx_i, {}_Ny_i)$ in global coordinates
Q – nodes for geometrical shape functions (marked by □)
$Q_i=(\xi_i, \eta_i)$ in local coordinates
$Q_i=(x_i, y_i)$ in global coordinates
b)
$P_7=Q_3$
P_8
$P_9=Q_4$
P_4
P_5
P_6
$P_1=Q_1$
P_2
$P_3=Q_2$
c)
η
ξ
(–1, 1)
(1, 1)
(–1, –1)
(1, –1)

Fig. 1. Non-curvilinear quadrilateral Lagrangian element: a) general notation, b) exemplary element of the 2nd order, c) placement in local coordinate system

Subject to the equations (14-18) the Jacobian matrix determinant equals:

$$\det \boldsymbol{J} = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} = \alpha_0\xi + \beta_0\eta + \chi, \tag{19}$$

where:

$$\alpha_0 = \frac{1}{8}\big((x_4 - x_3)(y_1 - y_2) + (x_2 - x_1)(y_4 - y_3)\big), \tag{20}$$

$$\beta_0 = \frac{1}{8}\big((x_2 - x_4)(y_1 - y_3) + (x_3 - x_1)(y_2 - y_4)\big), \tag{21}$$

$$\chi = \frac{1}{8}\big((x_1 - x_4)(y_2 - y_3) + (x_3 - x_2)(y_1 - y_4)\big). \tag{22}$$

The determinant det***J***, contrary to the case for triangular elements [3], is not a constant value in terms of the local coordinates. This is a serious drawback as therefore a non-constant term appears in the denominator of (11). In the case of (5) and (12) one can still derive exact formulae (as in the case of triangular Lagrange elements) and analytically perform the integration, however when considering term (11) the approach needs to be different.

Whether exact or approximate formulae will be used – the derivations are complicated and therefore the pre-assembly needs to be performed with the use of software that is capable of performing symbolic computations on multivariate polynomials. The authors use their own symbolic computation library written in C# (which will be explained in another paper). If one does not want to implement their own algorithms then commercial software like Mathematica [6] or Maple [7] can be used.

4. Exact formulae of the pre-assembly

This section displays how the formulae of the pre-assembly can be derived for the terms denoted by *K* and *d*. These terms do not contain polynomials in the denominator and negative exponentiations hence the well-known rule for multivariate polynomial integration can be applied:

$$k_j \in \mathbb{N}_0 \quad \to \quad \int \Big(\prod_i u_i^{k_i}\Big)\mathrm{d}u_j = \frac{1}{k_j+1} u_j^{k_j+1} \prod_{i \neq j} u_i^{k_i}, \tag{23}$$

where *u* with the appropriate indices are the polynomial variables.

When considering $K_{a,b}$ one can notice that generally:

$$N_a N_b = \sum_i v_i \xi^{k_i} \eta^{l_i}, \qquad v_i \in \mathbb{R};\ k_i, l_i \in \mathbb{N}_0 \tag{24}$$

therefore:

$$K_{a,b} = -\chi \int_{\xi=-1}^{1} \int_{\eta=-1}^{1} \left(\alpha \sum_i v_i \xi^{k_i+1} \eta^{l_i} + \beta \sum_i v_i \xi^{k_i} \eta^{l_i+1} + \sum_i v_i \xi^{k_i} \eta^{l_i} \right) \mathrm{d}\xi \mathrm{d}\eta, \tag{25}$$

hence generally the form of $K_{a,b}$ is:

$$K_{a,b} = \chi(c_{a,b;\alpha}\alpha + c_{a,b;\beta}\beta + c_{a,b;1}). \tag{26}$$

where:

$$\alpha = \frac{\alpha_0}{\chi}, \quad \beta = \frac{\beta_0}{\chi}. \tag{27}$$

The advantage of this form is that the *c* coefficients can be obtained globally for a non-curvilinear quadrilateral Lagrangian element of arbitrary order while only the α, β and χ need to be evaluated for each element using (20-22) and (27).

Hence an auxiliary data sheet for $K_{a,b}$ can be formulated in the form of a 3D structure (Figure 2). The structure does not have to be presented for $b > a$ as:

$$K_{a,b} = K_{b,a}. \tag{28}$$

a)

$$\left[\begin{array}{lllll}
\begin{bmatrix} c_{1,1;\alpha} & c_{1,1;\beta} & c_{1,1;1} \end{bmatrix} & & & & \\
\begin{bmatrix} c_{2,1;\alpha} & c_{2,1;\beta} & c_{2,1;1} \end{bmatrix} & \begin{bmatrix} c_{2,2;\alpha} & c_{2,2;\beta} & c_{2,2;1} \end{bmatrix} & & & \\
\begin{bmatrix} c_{3,1;\alpha} & c_{3,1;\beta} & c_{3,1;1} \end{bmatrix} & \begin{bmatrix} c_{3,2;\alpha} & c_{3,2;\beta} & c_{3,2;1} \end{bmatrix} & \begin{bmatrix} c_{3,3;\alpha} & c_{3,3;\beta} & c_{3,3;1} \end{bmatrix} & & \\
\quad\quad\vdots & \quad\quad\vdots & \quad\quad\vdots & & \\
\begin{bmatrix} c_{n,1;\alpha} & c_{n,1;\beta} & c_{n,1;1} \end{bmatrix} & \begin{bmatrix} c_{n,2;\alpha} & c_{n,2;\beta} & c_{n,2;1} \end{bmatrix} & \begin{bmatrix} c_{n,3;\alpha} & c_{n,3;\beta} & c_{n,3;1} \end{bmatrix} & \cdots & \begin{bmatrix} c_{n,n;\alpha} & c_{n,n;\beta} & c_{n,n;1} \end{bmatrix}
\end{array}\right]$$

b)

$$\left[\begin{array}{lll}
\begin{bmatrix} -0.8533 & -0.8533 & 1.1378 \end{bmatrix} & & \\
\begin{bmatrix} 0.2133 & 0.2844 & -0.2844 \end{bmatrix} & \begin{bmatrix} -0.8533 & 0 & 1.1378 \end{bmatrix} & \\
\begin{bmatrix} 0.2133 & 0 & -0.2844 \end{bmatrix} & \begin{bmatrix} 0.2133 & -0.2844 & -0.2844 \end{bmatrix} & \begin{bmatrix} -0.8533 & 0.8533 & 1.1378 \end{bmatrix}
\end{array}\right]$$

Fig. 2. The coefficients required for the evaluation of $K_{a,b}$ depicted in the form of a 3D structure: a) general display, b) exemplary values for an element of the 2nd order

At the element level – equation (26) relates to K_Ω (for $w = N_a$) in the following way:

$$K_\Omega\Big|_{w=N_a} = \sum_{b=1}^{m} \left(\gamma \frac{\partial A_b}{\partial t} + J_{\text{ext}\,b}\right) K_{a,b}. \tag{29}$$

The expression d_Ω is considered next. The manner in which d_Ω is handled for the a-th test function depends mostly on four overall questions:

a) is the element connected to a boundary of the studied domain?
b) does the node $({}_Nx_a,\ {}_Ny_a)$ belong to the boundary edge?
c) what type of boundary condition is imposed?
d) is the boundary condition homogeneous on the element's edge?

The term d_Ω is omitted completely unless both a) and b) are true. The rest depends on the combination of c) and d). Only the case for the homogeneous Neumann boundary condition is discussed in this paper. This case is the most convenient to handle since $\dfrac{\partial A}{\partial n}$ can be put in front of the integration:

$$d_\Omega = \frac{1}{\mu} \frac{\partial A}{\partial n} \oint_{\partial\Omega} w \, \mathrm{d}\Gamma. \tag{30}$$

The rest depends on which basis function one currently considers for w. Let us assume that ϑ represents a variable in the local coordinate system whose axis has the same direction as the boundary curve (Figure 3).

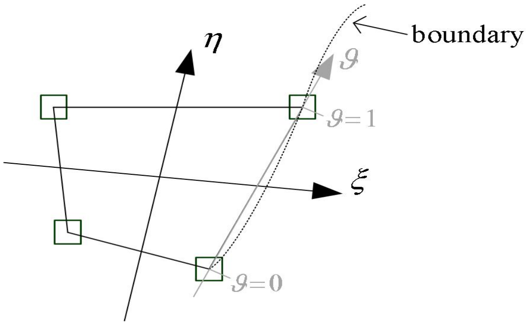


Fig. 3. Visualization of the auxiliary variable ϑ and its axis along the boundary

One can then build an auxiliary formula:

$$d_a = \int\limits_{\vartheta=-1}^{\vartheta=1} \prod_{\substack{i=1\\ i\neq a}}^{n} \frac{\vartheta-\vartheta_i}{\vartheta_a-\vartheta_i}\mathrm{d}\vartheta, \tag{31}$$

where n is the number of nodes on the element edge that represents the boundary fragment. At the element-level – equation (31) relates to d_Ω (for $w = N_a$) in the following way:

$$d_\Omega\big|_{w=N_a} = \Gamma_\mathrm{b}\frac{1}{\mu}\frac{\partial A}{\partial n}d_a, \tag{32}$$

where Γ_b is the length of the boundary fragment.

5. Aproximate formulae of the pre-assembly

This section displays how the formulae of the pre-assembly can be derived for the term denoted by Λ. It contains a non-constant expression (in terms of ξ and η) in the denominator, hence the integration cannot be performed with the use of (23). Due to (14) one can write (11) as:

$$\Lambda_{a,b} = -\int\limits_{\xi=-1}^{1}\int\limits_{\eta=-1}^{1}\left(\begin{array}{l}\left(\begin{array}{l}(\sum\limits_{l=1}^{4} y_l\frac{\partial G_l}{\partial\eta})\frac{\partial N_a}{\partial\xi} - \\ -(\sum\limits_{l=1}^{4} y_l\frac{\partial G_l}{\partial\xi})\frac{\partial N_a}{\partial\eta}\end{array}\right)\left(\begin{array}{l}(\sum\limits_{l=1}^{4} y_l\frac{\partial G_l}{\partial\eta})\frac{\partial N_b}{\partial\xi} - \\ -(\sum\limits_{l=1}^{4} y_l\frac{\partial G_l}{\partial\xi})\frac{\partial N_b}{\partial\eta}\end{array}\right) + \\ + \left(\begin{array}{l}-(\sum\limits_{l=1}^{4} x_l\frac{\partial G_l}{\partial\eta})\frac{\partial N_a}{\partial\xi} + \\ +(\sum\limits_{l=1}^{4} x_l\frac{\partial G_l}{\partial\xi})\frac{\partial N_a}{\partial\eta}\end{array}\right)\left(\begin{array}{l}-(\sum\limits_{l=1}^{4} x_l\frac{\partial G_l}{\partial\eta})\frac{\partial N_b}{\partial\xi} + \\ +(\sum\limits_{l=1}^{4} x_l\frac{\partial G_l}{\partial\xi})\frac{\partial N_b}{\partial\eta}\end{array}\right)\end{array}\right)\frac{1}{\det\boldsymbol{J}}\mathrm{d}\xi\mathrm{d}\eta. \tag{33}$$

Here an auxiliary notation is introduced:

$$Z_{a,l} = \frac{\partial G_l}{\partial \eta}\frac{\partial N_a}{\partial \xi} - \frac{\partial G_l}{\partial \xi}\frac{\partial N_a}{\partial \eta}, \tag{34}$$

hence:

$$\Lambda_{a,b} = -\sum_{l=1}^{4}\sum_{k=1}^{4}(y_k y_l + x_k x_l)\int\limits_{\xi=-1}^{1}\int\limits_{\eta=-1}^{1}\frac{Z_{a,l}Z_{b,k}}{\det \boldsymbol{J}}\mathrm{d}\xi\mathrm{d}\eta. \tag{35}$$

For further convenience, in order to reduce the number of added terms, one can write:

$$\Lambda_{a,b} = \sum_{l=1}^{4}\sum_{k=1}^{l}(y_k y_l + x_k x_l)\int\limits_{\xi=-1}^{1}\int\limits_{\eta=-1}^{1}\frac{Z_{a,b,l,k}}{\det \boldsymbol{J}}\mathrm{d}\xi\mathrm{d}\eta, \tag{36}$$

where:

$$Z_{a,b,l,k} = \begin{cases} -Z_{a,l}Z_{b,l} & l = k, \\ -Z_{a,l}Z_{b,k} - Z_{a,k}Z_{b,l} & l \neq k. \end{cases} \tag{37}$$

In order to reduce the number of symbolic variables – again the coefficient χ is put in front of the remaining part of the expression:

$$\frac{1}{\det \boldsymbol{J}} = \frac{1}{\chi}\frac{1}{(1+\alpha\xi+\beta\eta)}, \tag{38}$$

hence:

$$\Lambda_{a,b} = \frac{1}{\chi}\sum_{l=1}^{4}\sum_{k=1}^{l}(y_k y_l + x_k x_l)\int\limits_{\xi=-1}^{1}\int\limits_{\eta=-1}^{1}\frac{Z_{a,b,l,k}}{1+\alpha\xi+\beta\eta}\mathrm{d}\xi\mathrm{d}\eta, \tag{39}$$

and the expression under the integral is a function of ξ, η, α and β.

Obviously, for numerical reasons [8] one should avoid quadrilateral elements with angles above 180° (as then a singularity is encountered). Here one can notice an advantage of the form (38) as the denominator must only be considered as yielding positive values (obviously – because for $\xi = 0$, $\eta = 0$ it becomes 1 and if no internal angles above 180° are present in the element then the plane defined by $1+\alpha\xi+\beta\eta$ does not intersect with the area of the base square bounded by $\xi \in [-1, 1]$, $\eta \in [-1, 1]$).

The denominator is taken into account as:

$$r = 1+\alpha\xi+\beta\eta. \tag{40}$$

An attempt is made to approximate the reciprocal of r:

$$f(r) = \frac{1}{r}, \tag{41}$$

by means of a polynomial $\tilde{f}$ in order to allow integration by means of analytical formulae. At this stage it is worthwhile to observe how r intersects the (ξ, η) plane for typical element shapes (Figure 4 – next page).

In the analysis presented in Figure 4 the maximum and minimum values of r have been deliberately presented so that the interval, in which the function f is to be approximated, can be chosen. One can notice that even for quadrilaterals with angles close to 180° the value $r_{\min}$ does not reach below 0.1 hence in practice this can be the lower boundary of the approximation, while the upper boundary value could be anything above 2.

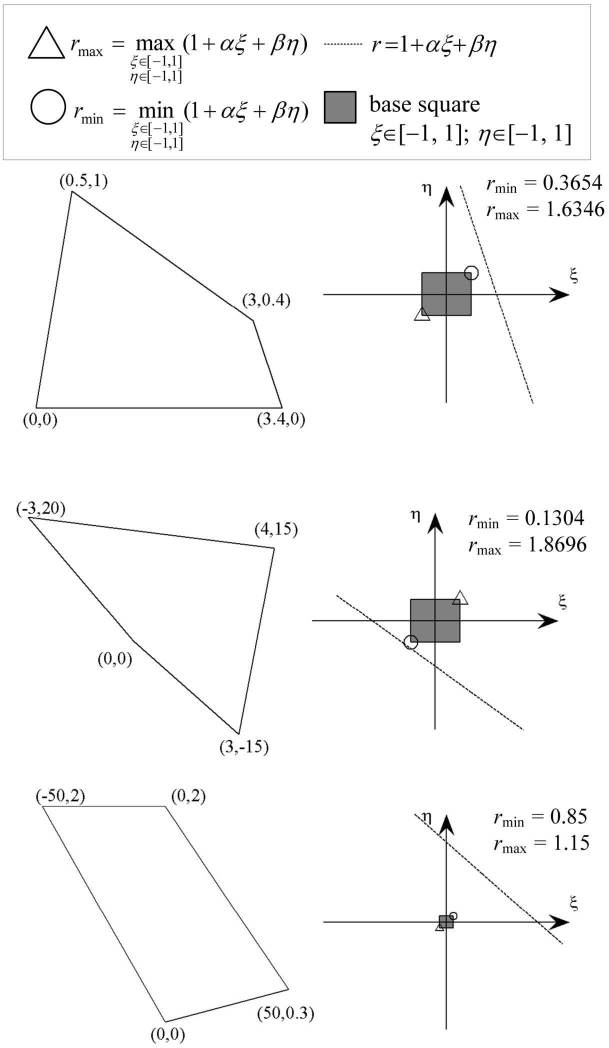

$r_{\max} = \max_{\substack{\xi\in[-1,1]\\ \eta\in[-1,1]}} (1+\alpha\xi+\beta\eta)$
$r=1+\alpha\xi+\beta\eta$
$r_{\min} = \min_{\substack{\xi\in[-1,1]\\ \eta\in[-1,1]}} (1+\alpha\xi+\beta\eta)$
base square
$\xi\in[-1, 1]$; $\eta\in[-1, 1]$
(0.5,1)
(3,0.4)
(0,0)
(3.4,0)
η
ξ
$r_{\min}$ = 0.3654
$r_{\max}$ = 1.6346
(-3,20)
(4,15)
(0,0)
(3,-15)
η
ξ
$r_{\min}$ = 0.1304
$r_{\max}$ = 1.8696
(-50,2)
(0,2)
(0,0)
(50,0.3)
η
ξ
$r_{\min}$ = 0.85
$r_{\max}$ = 1.15

Fig. 4. Examples of quadrilateral elements and their respective intersections of the *r* function with the ξ-η plane along with the minimum and maximum values of *r* within the base square

The approximating polynomial will be actually built with interpolation, however Chebyshev nodes must be used in order to avoid the Runge effect [9], which is a common occurrence for these types of functions. Next the polynomial $\widetilde{f}$ must assume the form dependent on the variables ξ and η. Subject to the approximation – equation (39) becomes:

$$\Lambda_{a,b} \approx \frac{1}{\chi} \sum_{l=1}^{4} \sum_{k=1}^{l} (y_k y_l + x_k x_l) \int_{\xi=-1}^{1} \int_{\eta=-1}^{1} Z_{a,b,l,k} \widetilde{f} \, \mathrm{d}\xi \mathrm{d}\eta, \tag{42}$$

where $\widetilde{f}$ initially assumes the form:

$$\widetilde{f} = v_0 + \sum_{i=1}^{n_\mathrm{I}-1} v_i r^i, \; v_i \in \mathbb{R}, \tag{43}$$

where n_I is the number of nodes used in the interpolation. When a substitution is performed according to (40) – a polynomial is obtained, which generally has the form:

$$\widetilde{f} = v_0 + \sum_{i=1}^{n_\mathrm{I}-1} v_i \left(\sum_{p+q+s=i} \frac{i!}{pqs} \xi^p \eta^q \alpha^p \beta^q \right), \tag{44}$$

following the multinomial theorem.

From the perspective of the variables α and β – $\widetilde{f}$ is a polynomial whose multipliers of each term $\alpha^p \beta^q$ could be stored in an upper-left triangular matrix (like depicted in Figure 5), where the polynomial follows the general formula:

$$\widetilde{f} = \sum_{p=0}^{n_\mathrm{I}-1} \sum_{q=0}^{n_\mathrm{I}-1-p} v_{p,q} \alpha^p \beta^q, \; v_{p,q} \in \mathbb{R}. \tag{45}$$

$$\begin{bmatrix} v_{0,0} & v_{0,1} & v_{0,2} & \cdots & v_{0,n_\mathrm{I}-3} & v_{0,n_\mathrm{I}-2} & v_{0,n_\mathrm{I}-1} \\ v_{1,0} & v_{1,1} & v_{1,2} & \cdots & v_{1,n_\mathrm{I}-3} & v_{1,n_\mathrm{I}-2} & \\ v_{1,0} & v_{1,1} & v_{1,2} & \cdots & v_{2,n_\mathrm{I}-3} & & \\ & \vdots & & & & & \\ v_{n_\mathrm{I}-3,0} & v_{n_\mathrm{I}-3,1} & v_{n_\mathrm{I}-3,2} & & & & \\ v_{n_\mathrm{I}-2,0} & v_{n_\mathrm{I}-2,1} & & & & & \\ v_{n_\mathrm{I}-1,0} & & & & & & \end{bmatrix}$$

Fig. 5. Illustration of the coefficients of *f* in an upper-left triangular matrix

Equation (39) takes the form:

$$\Lambda_{a,b} \approx \frac{1}{\chi} \sum_{l=1}^{4} \sum_{k=1}^{l} (y_k y_l + x_k x_l) \sum_{i=0}^{n_\mathrm{I}-1} \sum_{j=0}^{n_\mathrm{I}-1-i} \lambda_{a,b,l,k,i,j} \alpha^i \beta^j, \tag{46}$$

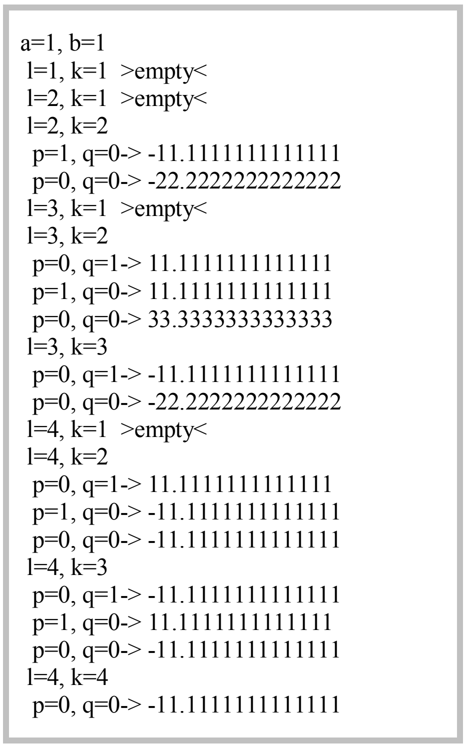
where according to the idea of the pre-assembly – the terms $\lambda_{a,b,l,k,i,j}$ can be prepared for a non-curvilinear quadrilateral Lagrangian element of a selected order and the equation above is to be used for the evaluation for each element in the mesh. $\Lambda_{a,b}$ can be then used in the equation for $\Lambda_\Omega\big|_{w=N_a}$:

$$\Lambda_\Omega\big|_{w=N_a} = \frac{1}{\mu} A_b \Lambda_{a,b}. \tag{47}$$

A final remark on the pre-assembly for Λ is that $\lambda_{a,b,l,k,i,j}$ needs to be only obtained for $b \le a$. A slight disadvantage is that the amount of values that one must store in a data sheet could be significant – for an element of order n_o (yielding $(n_o+1)^2$ basis functions) the amount of λ coefficients is:

$$s_\lambda = 2.5((n_o+1)^4 + (n_o+1)^2)(n_I{}^2 + n_I), \tag{48}$$

which means that for an element of the order n_o = 4 and n_I = 14 one needs to store maximally 341 250 values, which requires approximately 2.6 MB of memory. However, such an amount is insignificant in comparison to how much an FEM computation often requires. Also – the number of stored coefficients could be a lot smaller if occurrences where the multiplier of $\alpha^p\beta^q$ is 0 are not stored. A part of an exemplary data sheet of coefficients for a = 1, b = 1 for n_o = 1 is presented in Figure 6.



```
a=1, b=1
 l=1, k=1  >empty<
 l=2, k=1  >empty<
 l=2, k=2
  p=1, q=0-> -11.1111111111111
  p=0, q=0-> -22.2222222222222
 l=3, k=1  >empty<
 l=3, k=2
  p=0, q=1-> 11.1111111111111
  p=1, q=0-> 11.1111111111111
  p=0, q=0-> 33.3333333333333
 l=3, k=3
  p=0, q=1-> -11.1111111111111
  p=0, q=0-> -22.2222222222222
 l=4, k=1  >empty<
 l=4, k=2
  p=0, q=1-> 11.1111111111111
  p=1, q=0-> -11.1111111111111
  p=0, q=0-> -11.1111111111111
 l=4, k=3
  p=0, q=1-> -11.1111111111111
  p=1, q=0-> 11.1111111111111
  p=0, q=0-> -11.1111111111111
 l=4, k=4
  p=0, q=0-> -11.1111111111111
```

Fig. 6. Data sheet containing the λ coefficients for a = 1, b = 1 and n_o = 1, n_I = 2

5. Summary

The pre-assembly for selected element type, used for an FEM analysis, allows to apply relatively simple equations and prepared auxiliary data sheets to perform the assembly of the system of equations. This has been presented on the example of the weak formulation generally describing a linear transient magnetic field problem. For the described case, if the pre-assembly had been performed, one only needs to possess the data sheets, with the values of the d, c and λ coefficients, and apply the following equations to complete the assembly:

- (20-22) and (27) for the evaluation of α, β and χ,
- (32) for the d_{Ω} term,
- (26) and (29) for K_{Ω},
- (46) and (47) for Λ_{Ω}.

The main advantages that could be gained from applying the pre-assembly are:

- simplicity in the implementation of the auxiliary equations and data sheets when writing FEM codes,
- a potential decrease in the time used for the assembly of the equations (especially useful for time-dependent analyses, where the meshing is performed for each time-step),
- an assurance of an integration with good accuracy (based on a preceding approximation) of some expressions – like (11) (potentially – this could also be a great advantage in the future when the pre-assembly will be presented for curvilinear elements) as f is approximated a priori when one can globally check the accuracy of the approximation in a selected interval for r values.

The paper has shown the formula needed to perform the pre-assembly. Its usefulness and efficiency for selected problems will be discussed in future papers. Further research will concern the pre-assembly for 3D elements and curvilinear elements.

References

[1] Barański M., Demenko A., Łyskawiński W., Szeląg W.: Finite element analysis of transient electromagnetic-thermal phenomena in a squirrel cage motor. COMPEL, Vol. 30 No. 3 (2011).

[2] Karban P., Mach F., Kůs P., Pánek D., Doležel I.: Numerical solution of coupled problems using code Agros2D. Computing Volume 95, Issue 1 Supplement, pp. 381-408 May (2013).

[3] Sowa M., Typańska D.: Pre-assembly of FEM n-th order triangular elements. Pozn. Univ. Technol. Acad. J., Electr. Eng. No. 77, 133-140 (2014).

[4] Bolkowski S., Stabrowski M., Skoczylas J., Sroka J., Sikora J., Wincenciak S., Computer methods for the analysis of the electromagnetic field (in Polish). WNT, Warsaw (1993).

[5] Sikora J., Numerical methods of solving boundary value problems. Basics of the finite element method and the boundary element method (in Polish). Lublin University of Technology Publishing House (2012).

[6] www.wolfram.com/mathematica/

[7] www.maplesoft.com/products/maple/

[8] Freitag L.A., Plassmann P.E.: Local optimization-based untangling algorithms for quadrilateral meshes. Proceedings of the 10th International Meshing Roundtable, Newport Beach, CA, 397-406 (2001).

[9] Danalia I., Joly P., Kaber S.M., Postel M.: An Introduction to Scientific Computing. Twelve Computational Projects Solved with MATLAB. Springer Science+Business Media, LLC, New York (2007).