A combined implicit–explicit algorithm in time for non-linear finite element analysis

J. L. Curiel Sosa*,†, E. de Souza Neto‡ and D. R. J. Owen§

Civil and Computational Engineering Centre, University of Swansea, Singleton Park SA2 8PP, U.K.

SUMMARY

An algorithm combining the numerical execution in time of implicit and explicit methods of solution is presented in this paper. The algorithm swaps between both methods as required for the analysis. The whole mesh is solved for an unique method at once, i.e. there are no partitions of the mesh for separate implicit or explicit treatment of the solution. The combination is in-time, in such a manner that if the implicit method starts diverging the explicit one is initiated by appropriate conditions of transition. The formulation is presented first and its implementation is validated by the analysis of a key numerical example.

KEY WORDS: explicit–implicit; non-linear; finite element; large deformation; buckling; contact

INTRODUCTION

Some problems in finite element analysis may not be readily solved by implicit methods of solution such as contact between corner and curved concave surface [1–3]. In such cases, where a sharp-pointed edge contacts a concave surface, explicit (EXP) or direct integration of momentum equations is an alternative scheme of solution. However, the last would involve a high computational cost in those domains with a high number of elements (as its stability is conditioned by means of a critical time step).

At this stage, the idea of executing an implicit method until divergence arises and at that point starting to solve with an explicit method is highlighted. Moreover, the processing flow is returned back to the implicit scheme (IMP) when divergence origin has finished and the external loading is not totally applied.

*Correspondence to: J. L. Curiel Sosa, Civil and Computational Engineering Centre, University of Swansea, Singleton Park SA2 8PP, U.K.
†E-mail: cgeuriel@swansea.ac.uk
‡E-mail: e.desouzaneto@swansea.ac.uk
§E-mail: drj.owen@swansea.ac.uk

Copyright © 2005 John Wiley & Sons, Ltd.
The aim of this article is to illustrate the implicit/explicit algorithm in time without going into contact problems at this time. The solution of a part of the mesh with an implicit method and the other part with an explicit one is widely studied [4] and is not considered in this study. A description of the implicit and explicit schemes coded and their connection is presented below in order to illustrate the algorithm. A numerical example of buckling with geometric non-linearity is presented to validate the algorithm.

**IMPLICIT FORMULATION**

The solution of the weak form of momentum equations by Newton–Raphson method (NRM) is used as implicit method. In non-linear analysis, a convenient scheme to integrate the rate constitutive equation is required. Algorithm will depend on the type of material and whether finite strains are considered or not. The implicit algorithm is devoted to general case of path-dependent materials and large deformation analysis, taking into account that this is not the central issue in this study.

The implicit algorithm is based upon a pseudo-time discretization considering the transition of deformation between two time points. Thus, if a set of internal variables \( \alpha_n \) are given at \( t_n \) the deformation rate tensor \( \varepsilon(t_n+1) \) determine the stresses \( \sigma(t_n+1) \) and internal variables only through integration algorithm, Equations (1) and (2) (see Reference [5] for a description of the return mapping algorithm)

\[
\sigma(t_n+1) = \hat{\sigma}(\alpha_n, \varepsilon_{n+1}) \\
\alpha(t_n+1) = \hat{\alpha}(\alpha_n, \varepsilon_{n+1})
\]

After finite element discretization, the problem is committed to find displacements \( u_{n+1} \) at time \( t_{n+1} \), so that the incremental non-linear finite element equation (3) is satisfied

\[
R(u_{n+1}) = f^{\text{int}}(u_{n+1}) - f^{\text{ext}}_{n+1} = 0
\]

where internal and external forces vectors are obtained by

\[
f^{\text{int}}(u_{n+1}) = \sum_{e=1}^{n_{\text{elem}}} \left\{ \int_{\Omega(e)} B^T \sigma(\alpha_n, \varepsilon(u_{n+1})) \, dv \right\}
\]

\[
f^{\text{ext}}_{n+1} = \sum_{e=1}^{n_{\text{elem}}} \left\{ \int_{\Omega(e)} N^T b_{n+1} \, dv + \int_{\partial\Omega(e)} N^T q_{n+1} \, ds \right\}
\]

where \( N(\xi, \eta) \) are bilinear shape functions, \( b_{n+1} \) are the body forces, \( q_{n+1} \) the traction forces applied over the boundary of the body and \( B \) is the linear strains operator which has the next format (in plane stress/strain analysis) for the generic element \( e \) (the first subindex denotes

\*In general, path-dependent materials.
number of local node and the comma derivative)

\[
B = \begin{bmatrix}
N_{1,1}^{(e)} & 0 & N_{2,1}^{(e)} & 0 & \ldots & N_{n_{nodes},1}^{(e)} & 0 \\
0 & N_{1,2}^{(e)} & 0 & N_{2,2}^{(e)} & \ldots & 0 & N_{n_{nodes},2}^{(e)} \\
N_{1,2}^{(e)} & N_{1,1}^{(e)} & N_{2,2}^{(e)} & N_{2,1}^{(e)} & \ldots & N_{n_{nodes},2}^{(e)} & N_{n_{nodes},1}^{(e)}
\end{bmatrix}
\]

Equation (3) needs to be linearized in order to accomplish a numerical procedure (the integrals are approximated by Gaussian quadratures).

**SOLUTION TO THE INCREMENTAL PROBLEM**

NRM has been used in the solution of the equation due to its quadratic rates of convergence (3). Within elastic materials, solution is immediate as no integration algorithm is needed. The problem consists of solving the linearized version of Equation (3) for the incremental global displacement \(\delta u^{(k)}\) in each iteration

\[
K_T \delta u^{(k)} = -R^{(k-1)}(u_{n+1})
\]

where \(K_T\) is the global tangent stiffness matrix given as

\[
K_T = \frac{\partial R}{\partial u_{n+1}} \bigg|_{u_{n+1}^{(k-1)}}
\]

which is obtained by assembly of element stiffness matrices

\[
k_T^{(e)} = \int_{\Omega^{(e)}} B^T \hat{D} B \, dv
\]

where \(\hat{D}\) is the consistent tangent stiffness matrix [6]

\[
\hat{D} = \frac{\partial \hat{\sigma}}{\partial \hat{E}_{n+1}} \bigg|_{\hat{E}_{n+1}^{(k-1)}}
\]

The implicit scheme is exposed in compact form as follows (note at point (VII) that the flow is diverted to explicit if needed):

1. Initiate \((k = 0)\)

   \[
u_n^{(0)} = u_n
   \]

   \[
   R = f^{int}(u)_n - f^{ext}
   \]

2. For all elements, calculate consistent tangent stiffness matrix

   \[
   \hat{D} = \frac{\partial \hat{\sigma}}{\partial \hat{E}_{n+1}}
   \]
(III) Assemble stiffness matrices

\[ k_f^{(c)} = \sum_{j=1}^{n_{gauss}} \xi_j B_j^T \hat{D}_j B_j \]

where \( \xi_j \) is the weight associated to the Gauss point \( j \).

(IV) Increment iteration counter \((k = k + 1)\), assemble, solve the linearized equilibrium equation (13) and update stresses and internal variables:

\[ u_{n+1}^{(k)} = u_{n+1}^{(k-1)} + \delta u^{(k)} \]

\[ \varepsilon_{n+1}^{(k)} = B u_{n+1}^{(k)} \]

\[ \sigma_{n+1}^{(k)} = \hat{\sigma}(\mathbf{a}_n, \varepsilon_{n+1}^{(k)}) \]

\[ \mathbf{a}_{n+1}^{(k)} = \hat{\mathbf{a}}(\mathbf{a}_n, \varepsilon_{n+1}^{(k)}) \]

(V) Compute new internal forces at each element

\[ f_{\text{int}}^{(k)} = \sum_{j=1}^{n_{gauss}} \xi_j J_j B_j^T \sigma_{n+1,j}^{(k)} \]

(VI) Assemble element internal forces vector and update residual.

(VII) If iterations diverge then go to the EXP scheme (see Figure 1), else:

(a) If

\[ \frac{\| f_{\text{ext}} - f_{\text{int}} \|}{\| f_{\text{ext}} \|} \leq \xi \]

then solution for current external load is reached and values for this load are from last iteration \((\bullet)_{n+1} = (\bullet)^{(k)}_{n+1}\)

(b) else go to (II).

(VIII) If the total load is not completely applied, increment external load and go to (II). else exit.

The solution obtained from the last load increment (displacement and residual) that converged with IMP is taken as an initial condition for the EXP.

FORMULATION FOR FINITE STRAINS

In the case of finite elasticity, the material model is independent from the path and, hence, internal variables are not needed in the estimation of stresses. They can be evaluated without any numerical integration algorithm.

The integration algorithm for the general case of path-dependent materials is used to update stresses and other state variables with the addition of finite strains [7].
Figure 1. Combined in-time implicit–explicit algorithm.
The stresses are given by the algorithmic incremental constitutive function \( \sigma_{n+1} = \hat{\sigma}(\mathbf{u}_n, \mathbf{F}_{n+1}) \), being \( \mathbf{F}_{n+1} \) the deformation gradient at the end of the interval \([t_n, t_{n+1}]\).

Now, the force vectors are defined within the deformed configuration,

\[
\mathbf{f}_{\text{int}}(\mathbf{u}_{n+1}) = \bigwedge_{e=1}^{n_{\text{elem}}} \left\{ \int_{\varphi_{n+1}(\Omega^{(e)})} \mathbf{B}^T \sigma(\mathbf{u}_n, \mathbf{F}(\mathbf{u}_{n+1})) \, d\mathbf{v} \right\}
\]

\[
\mathbf{f}_{\text{ext}}(\mathbf{u}_{n+1}) = \bigwedge_{e=1}^{n_{\text{elem}}} \left\{ \int_{\varphi_{n+1}(\Omega^{(e)})} \mathbf{N}^T \mathbf{b}_{n+1} \, d\mathbf{v} + \int_{\partial\varphi_{n+1}(\Omega^{(e)})} \mathbf{N}^T \mathbf{q}_{n+1} \, d\mathbf{s} \right\}
\]

where \( \varphi_{n+1}(\Omega^{(e)}) \) is the current deformed domain. A generic iteration of NR involves solving the standard linear system

\[
\mathbf{K}_T \delta \mathbf{u}^{(k)} = -\mathbf{R}^{(k-1)}
\]

\( \mathbf{K}_T \) is computed as follows:

\[
\mathbf{K}_T = \bigwedge_{e=1}^{n_{\text{elem}}} \left\{ \int_{\varphi_{n+1}(\Omega^{(e)})} \mathbf{G}^T \mathbf{a} \mathbf{G} \, d\mathbf{v} \right\}
\]

where \( \mathbf{G} \), in plane stress/strain analysis, has the format

\[
\mathbf{G} = \begin{bmatrix}
N_{1,1}^{(e)} & 0 & N_{2,1}^{(e)} & 0 & \ldots & N_{m_{\text{node}},1}^{(e)} & 0 \\
0 & N_{1,1}^{(e)} & 0 & N_{2,1}^{(e)} & \ldots & 0 & N_{m_{\text{node}},1}^{(e)} \\
N_{1,2}^{(e)} & 0 & N_{2,2}^{(e)} & 0 & \ldots & N_{m_{\text{node}},2}^{(e)} & 0 \\
0 & N_{1,2}^{(e)} & 0 & N_{2,2}^{(e)} & \ldots & 0 & N_{m_{\text{node}},2}^{(e)}
\end{bmatrix}
\]

The matrix form of the fourth-order consistent spatial tangent modulus \( \mathbf{a} \) is written in cartesian components [7] as follows:

\[
a_{ijkl} = \frac{1}{J} \frac{\partial \mathbf{\tau}_{ij}}{\partial F_{km}} F_{lm} = \sigma_{ij} \delta_{jk}
\]

where \( \mathbf{\tau} \) is the Kirchoff stress tensor \((\mathbf{\tau} = \det[\mathbf{F}] \mathbf{\sigma})\). An equivalent incremental function is defined

\[
\mathbf{\tau}_{n+1} = \hat{\mathbf{\tau}}(\mathbf{u}_n, \mathbf{F}_{n+1}) = \det[\mathbf{F}_{n+1}] \hat{\sigma}(\mathbf{u}_n, \mathbf{F}_{n+1})
\]

The main modifications, to the algorithm given in the former section, are stated here:

(II) Computation of consistent spatial tangent modulus

\[
\hat{a}_{ijkl} = \frac{1}{J} \frac{\partial \hat{\mathbf{\tau}}_{ij}}{\partial F_{km}} F_{lm} = \sigma_{ij} \delta_{jk}
\]

(III) Assembly of element stiffness matrices

\[
\mathbf{k}_{\text{f}}^{(e)} = \sum_{i=1}^{n_{\text{max}}} \omega_i \mathbf{j}_i \mathbf{G}_i^T \mathbf{a} \mathbf{G}_i
\]
where \( \omega_i \) is the weight associated to generic gauss point \( i \) and \( j_i \) is the determinant of the Jacobian of the transformation evaluated in Gauss point \( i \).

(IV) Updating of deformation gradient

\[
F_{n+1}^{(k)} = (I - \nabla_x u_{n+1}^{(k)})^{-1}
\]

Use of constitutive integration algorithm to update the stress and other state variables

\[
\sigma_{n+1}^{(k)} = \hat{\sigma}(\sigma_n, F_{n+1}^{(k)}) \\
\alpha_{n+1}^{(k)} = \hat{\alpha}(\sigma_n, F_{n+1}^{(k)})
\]

**EXPLICIT FORMULATION**

The EXP execution is only activated in case of a divergence of the IMP method. Once the solution is achieved, the process is returned back to an IMP execution (see Figure 1) if the external loading is not totally applied.

The central differences method (CDM) has been chosen as explicit method by its simplicity. The global internal forces vector is given in Equation (4). CDM consists of integrating directly the spatially discretized dynamic equilibrium equation by central differences at time \( t^n \)

\[
M \ddot{u}(t_n) + C \dot{u}(t_n) + f^{int}(u_n) = f^{ext}(t_n)
\]  

where \( M \) is the mass matrix, \( C \) the damping matrix, and \( \ddot{u}(t_n), \dot{u}(t_n), u(t_n) \) are, accordingly, the acceleration, velocity and displacement vectors.

\[
\ddot{u}(t_{n-1/2}) = \frac{u(t_n) - u(t_{n-1})}{\Delta t_n}
\]

\[
\dot{u}_{t,n+1/2} = \left[ \frac{M_{ii}}{\Delta t_n} + \frac{C_{ii}}{2} \right]^{-1} \left[ \left[ \frac{M_{ii}}{\Delta t_n} - \frac{C_{ii}}{2} \right] \dot{u}_{t,n-1/2} + f^{int}(u(t_n)) - f^{ext}(t_n) \right]
\]

The following step is the calculation of displacements for next iteration and updating of stresses, internal variables and internal forces:

\[
u(t_{n+1}) = u(t_n) + \dot{u}_{t,n+1/2} \Delta t_{n+1}
\]

An adaptive time step is adopted. Thus, in each temporal iteration, a bigger step may be properly elected if the stability criterion allows it, saving computational time. A lumped mass matrix is elected by its simplicity. The diagonal matrix permits an uncoupled system of equations. It is evaluated in the undeformed configuration, prior to start iterations

\[
M = \sum_{\gamma=1}^{n_{elem}} \int_{\Delta \hat{\Omega}^\gamma} \rho_0 N^T_{(\gamma)} N_{(\gamma)} \, d\omega
\]

A technique of diagonalization has been used, which ensures mass conservation. This technique results in that the sum of masses in each point of the mesh is the total mass of the body. The
explicit integration can flow into strange and annoying oscillations. These can be effectively controlled by damping. Furthermore, in highly non-linear analysis it is convenient to use a damping which can vary with the stiffness [8]. The damping matrix is elected proportional to diagonal matrix $C_i$ to get computational advantage which leads to a system formed by uncoupled equations

$$ C = \alpha C_i $$

(21)

Mass matrix proportional damping is used [9] $C_i = M$. In addition, Munjiza’s proposal [10], which includes stiffness and mass relationship to obtain damping, has been used in the analysis that requires to damp a wide range of frequencies

$$ C_i = M(M^{-1}K)^m $$

(22)

$$ C_{ii} = 2 \omega_i \xi C_{ii} $$

(23)

Frequencies are calculated as indicated in Equation (28) [11].

**CRITICAL TIME STEP**

The time step for the CDM must be less than a specific value to guarantee the stability of the scheme and therefore its convergence. This value is bounded by natural frequencies and damping ratio relationship as stated in Equation (24)

$$ \Delta t \leq \min \left\{ \frac{2}{\omega_i} \left( -\xi_i + \sqrt{1 + \xi_i^2} \right) \right\} $$

(24)

where $\omega_i$ are the natural frequencies and $\xi_i$ the fraction of critical damping at each node $i$ of the mesh. This inequality is satisfied if the maximum frequency in the mesh is elected. The maximum frequency can be calculated knowing the maximum eigenvalue of the mesh as

$$ \omega_{max} = \sqrt{\lambda_{max}^{(mesh)}}. $$

Then, the maximum eigenvalue can be bounded by the maximum element eigenvalue $\lambda_{max}^{(mesh)} \leq \lambda_{max}^{(e)}$ [12].

Adaptive time step has been used in the programming and leads to save time of execution. The critical time step is elected as

$$ \Delta t(t_{n+1}) = \frac{2}{\max \{\omega_i(t_n)\}} $$

(25)

The natural frequencies are determined from the homogeneous problem, Equation (26). Its analytical solution is in the form $u(t) = iue^{-j\omega t}$ ($j = \sqrt{-1}$) and introducing this at Equation (26) renders an eigenvalue problem

$$ \mathbf{M}\ddot{u} + \mathbf{K}u = 0 $$

(26)

which gives next

$$ -\omega^2 \mathbf{M}(\cos(\omega t) - j \sin(\omega t)) + \mathbf{K}(\cos(\omega t) - j \sin(\omega t)) = 0 $$
which leads to the classical eigenvalue problem

$$| - \omega^2 M + K | = 0$$

where the \( \omega^2 \) are the eigenvalues of the system that provide the natural frequencies for each node and degree of freedom.

The stiffness matrix is never computed and an approximation is made instead

$$K_0(t_n) \approx \frac{f_{\text{int}}^n(u_n) - f_{\text{int}}^{n-1}(u_{n-1})}{\dot{u}_i^{n-1/2} \Delta t_n}$$

by which the stiffness matrix becomes diagonal. The natural frequencies may be easily calculated from Equation (28)

$$\omega_i(t_n) = \sqrt{\frac{K_0(t_n)}{M_{ii}}}$$

and the critical time step is computed as stated in Equation (25).

**TRANSITION IMPLICIT/EXPLICIT**

The divergence criterion (to give up IMP) is based upon the relative residual norm, expression (10). Upon failure of convergence within an incremental loading step, the last converged iteration at IMP corresponding to solution to the previous external loading step (displacement \( \tilde{u} \)) is transferred to EXP as an initial condition. Thus, the final internal forces and displacements from IMP are used to determine initial accelerations and velocities for EXP,

$$f_{\text{int}}(\tilde{u}) \rightarrow f_{\text{int}}(t = 0)$$

$$\tilde{u} \rightarrow u(t = 0)$$

Initial velocities for EXP are approached as follows:

$$\ddot{u}_i(0) = \frac{f_{\text{int}}^n - f_{\text{int}}^{n-1}(0)}{M_{ii}}$$

$$\dot{u}_i(0) = \ddot{u}_i(0) \Delta t(0) + \dot{u}_i^-(0)$$

$$\dot{u}_i^-(0) = 0.0$$

A general diagram of IMP/EXP scheme is represented in Figure 1.

**NUMERICAL EXAMPLE**

The buckling of an aluminium alloy (Young modulus \( E = 6.895 \times 10^4 \) MPa, Poisson ratio \( \nu = 0.34 \) and density \( \rho = 2700 \) kg/m\(^3\)) arch is presented. An external point load (up to a
Figure 2. Geometry of the fixed-at-edges arch.

Table I. Relative residual norm (%) (error (%) in the table) and maximum residual. Vertical internal force (N) and displacement (m) at central node of the arch \( (k_{cn}) \). In the second column, the iteration number \( (i) \) is shown for IMP sub-algorithm and the time \( (t_n) \) for EXP sub-algorithm.

<table>
<thead>
<tr>
<th>Sub-algor.</th>
<th>( i/t_n )</th>
<th>Error (%)</th>
<th>Max. Resid.</th>
<th>( f^\text{int}(k_{cn}) )</th>
<th>( u_y(k_{cn}) )</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMP</td>
<td>0</td>
<td>3118.820</td>
<td>197857.00</td>
<td>0.0</td>
<td>0.0</td>
<td>start.</td>
</tr>
<tr>
<td>IMP</td>
<td>1</td>
<td>59.346</td>
<td>3767.75</td>
<td>-7220.6</td>
<td>-0.061</td>
<td>—</td>
</tr>
<tr>
<td>IMP</td>
<td>2</td>
<td>3116.590</td>
<td>202416.00</td>
<td>-4015.2</td>
<td>-0.074</td>
<td>diverg.</td>
</tr>
<tr>
<td>EXP</td>
<td>0.0001</td>
<td>3116.590</td>
<td>202416.00</td>
<td>-4015.2</td>
<td>-0.074</td>
<td>oscillat.</td>
</tr>
<tr>
<td>EXP</td>
<td>0.001</td>
<td>64.204</td>
<td>785.12</td>
<td>-4389.4</td>
<td>-0.075</td>
<td>oscillat.</td>
</tr>
<tr>
<td>EXP</td>
<td>0.1</td>
<td>2.386</td>
<td>336.19</td>
<td>-3971.2</td>
<td>-0.466</td>
<td>oscillat.</td>
</tr>
<tr>
<td>EXP</td>
<td>0.5</td>
<td>0.092</td>
<td>29.74</td>
<td>-3998.5</td>
<td>-1.193</td>
<td>converg.</td>
</tr>
<tr>
<td>EXP</td>
<td>0.641</td>
<td>0.033</td>
<td>9.98</td>
<td>-3999.3</td>
<td>-1.17</td>
<td>converg.</td>
</tr>
<tr>
<td>EXP</td>
<td>1.0</td>
<td>0.007</td>
<td>2.016</td>
<td>-4001.27</td>
<td>-1.17</td>
<td>converg.</td>
</tr>
<tr>
<td>EXP</td>
<td>1.5</td>
<td>( 10^{-5} )</td>
<td>0.293</td>
<td>-4000.16</td>
<td>-1.17</td>
<td>converg.</td>
</tr>
<tr>
<td>EXP</td>
<td>2.0</td>
<td>( 4.4 \times 10^{-5} )</td>
<td>0.017</td>
<td>-4000.01</td>
<td>-1.17</td>
<td>solution</td>
</tr>
</tbody>
</table>

magnitude 4000 N that causes the snap-through of the arch) is applied in the middle of the arch as shown in Figure 2. Other geometric values are cross-section area \( A = 25.806 \times 10^{-4} \) m\(^2\), inertia moment \( I = 5.54 \times 10^{-7} \) m\(^4\), thickness \( t = 0.0508 \) m, and arch radius \( R = 5.08 \) m. References [13] or [14] simulated it in the pre-buckling behaviour. A further development was performed by Pin and Trahair [15] carrying out simulations after buckling point.

In the simulations performed, IMP began to diverge at deflection \( |\delta| = 0.074 \) m (\( |\delta_{\text{crit}}| = 0.076 \) m which corresponds to nodal internal force at vertical direction \( f^\text{int}_{\text{central node, } y} = 2781.917 \) N) iteration 2 in Table I, and, hence, the EXP is automatically initiated (indicated at point (3), Figure 3).

The evolution of the explicit iterations may be observed in Figure 4. Evolution of the arch for different external loadings is shown in Figures 5 and 6.
CONCLUSION

An algorithm that considers the combination in-time of two general strategies of solution (implicit and explicit) has been presented, in particular, for the analysis of large deformations.
Figure 5. Deformed configurations for different external loadings at midpoint. Nodal internal force and deflection at central node in brackets.

Figure 6. Evolution of the arch for different external loadings.
where geometrically non-linear snap-through buckling results in divergence of the implicit method used. Practical applications of this algorithm might also include the solution of contact problems between complex geometries such as sharp-pointed edges and curved concave surfaces or the solution to snap-back problems.

REFERENCES