# ابداع جدید در روش انتگرال‌گیری گام‌به‌گام برای حل مسائل میدان گذرا

**سیامک سلیمانی شیشوان [۱]، اسدا... نورزاد [۲]**

**دانشکده‌ی عمران ـ پردیس دانشکده‌های فنی ـ دانشگاه تهران**

Email: sshishvan@ut.ac.ir#

**خلاصه**

مدلهای ریاضی پدیده‌های فیزیکی بر حسب قوانین اساسی به معادلات دیفرانسیلی ختم می‌گردند. این معادلات در مسائل میدان گذرا از مرتبه‌ی اول در زمان می‌باشند و به مساله مقدار مرزی معروف هستند. حل این نوع مساله به روش عددی با گسسته سازی زمانی همراه با عملیات گام به گام صورت می‌گیرد. این نوع روشها با توجه به محدودیت اندازه‌ی بزرگی گام زمانی احتیاج به زمان زیادی جهت تحلیل دارند. محققان زیادی با ارائه‌ی الگوریتمهای مختلف سعی در بزرگ کردن گامهای زمانی داشته‌اند. در این تحقیق روشی نوین برای انتگرالگیری گام به گام با گامهای زمانی خیلی بزرگ با استفاده از بازتولید توابع ارائه می‌شود. در روش بازتولید، توابع شکل مورد نیاز با توجه به اندازه‌ی گام زمانی تولید می‌شوند، بنابراین در صورت استفاده از چنین توابع شکلی در انتگرالگیری گام به گام، می‌توان با گامهای زمانی مطلوب به نتایج دقیق دست یافت. در الگوریتم پیشنهادی برای ارضای معادله‌ی حاکم از روش همجایگی استفاده شده است. بدلیل محدودیت قضیه‌ی نمونه‌برداری در فواصل نمونه‌ها برای برازش منحنی دقیق به نقاط جواب، استفاده از الگوریتمهای مرسوم مرتبه بالا با گامهای زمانی بزرگ مقدور نمی‌باشد. در الگوریتم پیشنهادی با برآورد جواب دقیق بین‌گامی، این مشکل رفع شده است. در نقاط همجایگی مقادیر بارگذاری به صورت دلخواه می‌توانند تعریف شوند که ساده‌ترین شکل آن درونیابی خطی بین مقادیر دو انتهای گام زمانی است. بعلاوه در این الگوریتم می‌توان گامهای زمانی حل را متفاوت از گامهای زمانی تابع بار در نظر گرفت. نتایج الگوریتم پیشنهادی در سیستمهای یک درجه و چند درجه آزادی با حل مثالهای مختلف با بارگذاری‌های متفاوت مورد بررسی قرارگرفته است. در مقایسه با گامهای زمانی الگوریتمهای مرسوم برای رسیدن به جواب دقیق، می‌توان گامهای زمانی الگوریتم پیشنهادی را اَبَر گام معرفی نمود.

**کلمات کلیدی: انتگرال‌گیری گام‌به‌گام ، بازتولید توابع ، مسائل میدان گذرا ، روش همجایگی**

## مقدمه

هر مسئله یا پدیده‌ی فیزیکی با در نظر گرفتن یکسری فرضیات و اصول به صورت مدل ریاضی قابل بیان است. این مدل ریاضی می تواند به صورت تغییراتی و یا دیفرانسیلی بیان گردد. دسته‌ی بزرگی از مسایل فیزیکی تحت عنوان مسائل میدان گذرا مانند مسئله‌ی انتقال حرارت، نفوذ سیال شناخته می‌شوندکه در این مسایل، این مدل به صورت یک معادله‌ی دیفرانسیل مرتبه‌ی اول در زمان می باشد.

به دلیل پیچیدگی سیستمها و همچنین توابع تحریک در این مسائل معمولاً حل‌های بسته به سادگی قابل حصول نیست و بنابراین لازم است تا از روشهای عددی استفاده گردد. یکی از مشکلاتی که در پیش روی محققان قرار داشت این بود که اعمال پیچیدگی های مدلهای فیزیکی در مدلهای ریاضی به خاطر یکسری محدودیـتها ( مانند حافظه‌ی محدود کامپیوتر و سرعت پردازشگرها ) موجب کند شدن عملیات محاسباتی و حتی در مواردی عدم انجام آن می‌شد. این مشکلات باعث می شد که محققان برای بهینه نمودن الگوریتمها از لحاظ حافظه‌ی مورد لزوم و زمان انجام پردازش اقدامات متعددی انجام دهند و حتی الگوریتمهای متنوع جدیدی را پیشنهاد نمایند.در موارد زیادی نیز لازم است که با حجم کم عملیات به مراتب بالای دقت دست یابیم و این جز با درنظر گرفتن توابع تقریب با مرتبه‌ی بالا میسر نیست.

---

[۱] دانشجوی دکتری سازه

[۲] استادیار



Computer For Civil Software Engineering Group : www.CCSofts.com , www.CompCivil.com

در مورد مسائلی که با آن سر و کار داریم دو حوزه‌ی مکان و زمان معمولاً مستقلاً گسسته سازی می شوند بطوریکه پس از گسسته سازی در مکان به یکسری معادلات درگیر وابسته به زمان می‌رسیم که در مسائل میدان گذرا به صورت معادله‌ی دیفرانسیل مرتبه‌ی اول در زمان می‌باشند:

$$\mathbf{C}\dot{\mathbf{u}}(t)+\mathbf{K}\mathbf{u}(t)=\mathbf{f}(t) \qquad (1)$$

بطوریکه بردار $\mathbf{u}$ وابسته به زمان می‌باشد و ( $\dot{}$ ) مشتق نسبت به زمان را نشان می‌دهد. با توجه به ویژگی حوزه‌ی زمان مجبوریم که آن را در یک فاصله زمانی محدود $\Delta t$ و رابطه‌ی جواب در لحظه‌ی $t_n$ نسبت به لحظه‌ی $t_{n+1}=t_n+\Delta t$ خلاصه نماییم، که این رابطه به رابطه‌ی بازگشتی موسوم است. به عبارت دیگر گسسته سازی در این حوزه بایستی همراه با عملیات گام به گام صورت گیرد.

با توجه به گستردگی و تنوع الگوریتمهای گام به گام موجود که معمولاً به نام اشخاصی که این الگوریتمها را ارائه داده‌اند در ادبیات فنی یافت می شوند، سعی می شود به صورت کلی به این الگوریتمها اشاره شود [۱].

الگوریتمها را به طرق مختلف می توان دسته بندی نمود به عنوان نمونه آنها را می توان براساس دقت ، براساس پایداری و نحوه ارضاء شرایط مرزی، بر اساس میرایی الگوریتمی و غیره دسته بندی نمود.

یکی از این روشها الگوریتمهای تک‌مرحله‌ای می‌باشند که برای مسائل خطی مرتبه‌ی اول در حالت عمومی بسیار کارآ می‌باشند [۲]، الگوریتمهای تک‌مرحله‌ای به فرمهای مختلفی فرمول بندی می‌شوند که در این زمینه می‌توان به استفاده از روش باقیمانده‌ی وزنی (الگوریتم $\theta$ ) و استفاده از روش همجایگی سری های تیلور اشاره نمود. الگوریتم کرنک-نیکلسون ( $\theta=1/2$ ) یکی از الگوریتمهای تک‌مرحله‌ای بسیار متداول در کاربردهای مهندسی است و یک الگوریتم به صورت غیر مشروط پایدار می‌باشد[۲].از دیگر روشها، الگوریتمهای بازگشتی چندگامی عمومی را می‌توان نام برد که در این الگوریتمها مقدار جواب به مقادیر جواب در چند گام قبلی مرتبط می‌شود و در نتیجه دقت بالاتر می‌رود؛ این الگوریتمها در مسائل عملی کمتر مورد استفاده قرار می‌گیرند ولی ایده استخراج این روشها می‌تواند در نوع خود جالب باشد[۲]. یکی دیگر از روشها، روش زیرگام‌بندی با گام‌های زمانی مختلط می‌باشد. فانگ توانست با استفاده از روش زیرگامی برای معادله‌ی مرتبه‌ی اول، الگوریتم خوب و مناسبی که به صورت غیر مشروط پایدار بوده و دارای میرائی عددی قابل کنترل است، ارائه دهد. وی الگوریتم $\theta$ را به عنوان مبنای الگوریتمی استفاده می‌کند و با ترکیب خطی نتایج زیرگامی مختلط با پارامترهای وزن مربوطه به دقت بالا می‌رسد. همچنین با وارد کردن شعاع طیفی نهایی به عنوان پارامتر الگوریتمی، میرائی عددی الگوریتم را کنترل می‌نماید[۳و۴]. دسته‌ی دیگر، الگوریتمهای با دقت مرتبه بالا با استفاده از پارامترهای وزنی می‌باشند. فانگ برای رسیدن به دقت مرتبه بالا و داشتن خواص مطلوب الگوریتمی، پیشنهاد می‌کند که در چندجمله‌ای مفروض به عنوان جواب تقریبی، ضرایب جملات غیر از جمله‌ی آخر را نیز مجهول در نظر بگیریم. ایشان برای حل از روش باقیمانده‌ی وزنی[۵] ویا از روش همجایگی [۶] استفاده نموده و با در نظر گرفتن یک چندجمله‌ای درجه n با n ضریب نامعین، الگوریتم با دقت مرتبه بالا به صورت غیر مشروط پایدار را پایه‌ریزی نموده است. همچنین فانگ بر اساس روش شبه حداقل مربعات به الگوریتم با دقت مرتبه بالا و به صورت غیر مشروط پایدار دست یافت [۷]. هدف کلی در این الگوریتمها تقریب تابع جواب در فاصله‌ی گام زمانی می باشد و هرچه مرتبه‌ی دقت این تقریبها بالاتر رود گامهای زمانی حل بزرگتر می‌شود، و با بکاربردن تدابیر خاصی مشخصات الگوریتم به صورت کنترل شده و پارامتری بیان می‌شود. در این تحقیق یک ابداع جدید در روش انتگرال‌گیری گام‌به‌گام با استفاده از بازتولید توابع ارائه شده [۱] و با حل مثالهای عددی برای سیستم‌های یکدرجه و چند درجه آزادی کارایی الگوریتم پیشنهادی تایید گردیده است.

### توابع انترپلاسیون

با استفاده از پردازش سیگنالها می‌توان یک تابع را در یک محدوده‌ی مورد نظر، باز تولید نمود [۸]، از آنجا که در نظر داریم از یک مرتبه توصیف پاسخ استفاده کنیم معادله‌ی باز تولید تابع u(t) به فرم زیر می‌باشد :

$$u^a(t)=\int_\Omega \bar{\phi}(t-\tau)u(\tau)d\tau \qquad (2)$$

که در آن $\bar{\phi}(t-\tau)$ ، تابع ویندوی تصحیح شده (تابع انترپلاسیون) می‌باشد وآن را به صورت زیر می‌توان نوشت :

$$\bar{\phi}(t-\tau)=C(t;t-\tau)\phi(t-\tau) \qquad (3)$$

که در آن $\phi(t-\tau)$ تابع ویندو یا تابع هسته‌ی پایه‌ای می‌باشد که از ترکیب خطی چندجمله‌ای‌های پایه تشکیل می‌گردد و $C(t;t-\tau)$ تابع تصحیح است که با ارضاء شرایط بازتولید بدست می‌آید.

بسط تیلور تابع $u(\tau)$ را به صورت زیر می‌توان نوشت:

$$u(\tau)=u(t)+(\tau-t)\left.\frac{\partial u(\tau)}{\partial\tau}\right|_{\tau=t}+\frac{(\tau-t)^2}{2!}\left.\frac{\partial^2 u(\tau)}{\partial\tau^2}\right|_{\tau=t}+\cdots \qquad (4)$$

انتخاب تعداد جملات بسط تیلور تابع $u(\tau)$ ، ارتباط مستقیم با فاصله‌ی زمانی‌ایی دارد که می‌خواهیم جواب را در آن فاصله بازتولید کنیم یعنی با افزایش تعداد جملات بسط تیلور تابع $u(\tau)$ می‌توانیم فاصله‌ی گام زمانی را بیشتر کنیم. تابع تصحیح را میتوان به فرم زیر نوشت:



Computer For Civil Software Engineering Group : www.CCSofts.com , www.CompCivil.com

$$C(t;t-\tau)=\mathbf{P}^T(t-\tau)\cdot\mathbf{b}(t)$$
$$\mathbf{P}^T(t-\tau)=[1,(t-\tau),\cdots,(t-\tau)^n] \qquad (5)$$
$$\mathbf{b}^T(t)=[b_0(t),b_1(t),\cdots,b_n(t)]$$

با توجه به تعداد جملاتی که در بسط به سری تیلور تابع $u(\tau)$ انتخاب می‌نماییم، n در تابع تصحیح انتخاب می‌شود و $b_i(t)$ ضرایب تابع تصحیح می‌باشند و باید تعیین شوند. برای اینکه با نحوه‌ی محاسبه‌ی تابع تصحیح آشنا شویم در زیر برای حالتی که از سه جمله‌ی بسط به سری تیلور تابع $u(\tau)$ استفاده کنیم، تابع تصحیح را بدست می‌آوریم؛ بنابراین $u(\tau)$ را مطابق رابطه‌ی زیر در نظر می‌گیریم:

$$u(\tau)=u(t)-(t-\tau)\dot{u}(t)+\frac{(t-\tau)^2}{2!}\ddot{u}(t) \qquad (6)$$

بنابراین تابع تصحیح را به فرم زیر باید در نظر گرفت:

$$C(t;t-\tau)=b_0(t)+(t-\tau)b_1(t)+(t-\tau)^2b_2(t) \qquad (7)$$

با جایگذاری $u(\tau)$ از رابطه‌ی (۶) و تابع ویندوی تصحیح شده از روابط(۳) و (۷) در رابطه‌ی (۲)، معادله‌ی بازتولید به فرم زیر درمی‌آید:

$$u^a(t)=\int_\Omega[b_0(t)+(t-\tau)b_1(t)+(t-\tau)^2b_2(t)][u(t)-(t-\tau)\dot{u}(t)+\frac{(t-\tau)^2}{2!}\ddot{u}(t)]\phi(t-\tau)\,d\tau \qquad (8)$$

با استفاده از تعریف ممان مطابق رابطه‌ی زیر:

$$m_i=\int_\Omega(t-\tau)^i\phi(t-\tau)d\tau \qquad (9)$$

رابطه‌ی (۸) به شکل زیر ساده می‌شود:

$$u^a(t)=\overline{m}_0(t)u(t)-\overline{m}_1(t)\dot{u}(t)+\frac{1}{2}\overline{m}_2(t)\ddot{u}(t) \qquad (10)$$

برای باز تولید دقیق تابع $u(t)$ یعنی $u^a(t)=u(t)$ ، در رابطه‌ی (۱۰) باید شرایط زیر ارضاء شوند:

$$\begin{cases}\overline{m}_0(t)=b_0(t)m_0+b_1(t)m_1+b_2(t)m_2=1\\ \overline{m}_1(t)=b_0(t)m_1+b_1(t)m_2+b_2(t)m_3=0\\ \overline{m}_2(t)=b_0(t)m_2+b_1(t)m_3+b_2(t)m_4=0\end{cases}\Rightarrow\begin{bmatrix}m_0&m_1&m_2\\m_1&m_2&m_3\\m_2&m_3&m_4\end{bmatrix}\cdot\begin{Bmatrix}b_0\\b_1\\b_2\end{Bmatrix}=\begin{Bmatrix}1\\0\\0\end{Bmatrix} \qquad (11)$$

از حل رابطه‌ی فوق ضرایب تابع تصحیح بدست می‌آیند بنابراین تابع ویندوی تصحیح شده بدست می‌آید که به عنوان یک تابع انترپلاسیون می‌تواند تابع $u(t)$ را بازتولید نماید. با استفاده از این تکنیک می‌توان مشتقات $u(t)$ را نیز بازتولید نمود، برای مشتق مرتبه‌ی اول رابطه‌ی بازتولیدرا می‌نویسیم سپس با استفاده از بسط تیلور تابع $u(\tau)$ از رابطه‌ی (۶) و تابع ویندوی تصحیح شده از روابط (۳) و (۷) ،رابطه‌ی بازتولید مشتق مرتبه‌ی اول تابع عبارت خواهد شد با:

$$\frac{\partial u^a(t)}{\partial t}=\int_\Omega\frac{\partial}{\partial t}\overline{\phi}(t-\tau)u(\tau)d\tau=\overline{m}_{0,t}(t)u(t)-(\overline{m}_{1,t}(t)-\overline{m}_0(t))\dot{u}(t)+\frac{1}{2}(\overline{m}_{2,t}(t)-2\overline{m}_1(t))\ddot{u}(t) \qquad (12)$$

برای باز تولید دقیق مشتق تابع $u(t)$ یعنی $\frac{\partial\dot{u}^a(t)}{\partial t}=\dot{u}(t)$ ، در رابطه‌ی (۱۲) بایستی شرایط زیر ارضاء شوند:

$$\begin{cases}\overline{m}_{0,t}(t)=0\\ \overline{m}_{1,t}(t)-\overline{m}_0(t)=-1\\ \overline{m}_{2,t}(t)-2\overline{m}_1(t)=0\end{cases}\Rightarrow\begin{bmatrix}m_0&m_1&m_2\\m_1&m_2&m_3\\m_2&m_3&m_4\end{bmatrix}_{,t}\cdot\begin{Bmatrix}b_0\\b_1\\b_2\end{Bmatrix}+\begin{bmatrix}m_0&m_1&m_2\\m_1&m_2&m_3\\m_2&m_3&m_4\end{bmatrix}\cdot\begin{Bmatrix}b_{0,t}\\b_{1,t}\\b_{2,t}\end{Bmatrix}=\begin{Bmatrix}0\\0\\0\end{Bmatrix} \qquad (13)$$

از حل معادله فوق ضرایب $b_{i,t}$ بدست می‌آیند. ملاحظه می‌شود که با استفاده از روش باز تولید تابع می‌توان هر مرتبه‌ی مشتق تابع را بازتولید نمود و ضرایب تصحیح برای بازتولید تابع و مشتقات تابع بدست می‌آیند که با انتخاب تعداد جملات بسط تیلور تابع $u(\tau)$ و متناسب با آن تابع تصحیح، می‌توانیم توابع انترپلاسیون مناسب برای باز تولید تابع را بدست آوریم.

### فرم گسسته‌ی بازتولید تابع

روش همجایگی یکی از روشهای حل معادلات می‌باشد، کارایی این روش برای بازتولید توابع در مرجع [۹] نشان داده شده است. اگر محدوده‌ی $\Omega$ در رابطه‌ی باز تولید، از NP نقطه‌ی مجزا(نقاط همجایگی) تشکیل شده باشد فرم گسسته‌ی رابطه‌ی (۲) به صورت زیر خواهد شد:



Computer For Civil Software Engineering Group : www.CCSofts.com , www.CompCivil.com

$$u^a(t)=\sum_{j=1}^{NP}\bar{\phi}(t-t_j)u(t_j)\Delta t_j$$
$$\bar{\phi}(t-t_j)=C(t;t-t_j)\phi(t-t_j)\quad;\quad C(t;t-t_j)=\mathbf{P}^T(t-t_j)\cdot\mathbf{b}(t) \tag{14}$$

رابطه‌ی (۱۴) را به شکل زیر بازنویسی می‌کنیم:

$$u^a(t)=\sum_{j=1}^{NP}N_j(t)u_j\quad;\quad u_j=u(t_j)\quad;\quad N_j(t)=C(t;t-t_j)\phi(t-t_j)\Delta t_j \tag{15}$$

که در آن $N_j(t)$ تابع انترپلاسیون یا تابع شکل می‌باشد و مشتقات آن به صورت زیر هستند:

$$N_{j,t}=(C_{,t}\,\phi+C\,\phi_{,t})\Delta t_j\quad;\quad N_{j,tt}=(C_{,tt}\,\phi+2C_{,t}\,\phi_{,t}+C\,\phi_{,tt})\Delta t_j \tag{16}$$

**حل معادله‌ی مرتبه‌ی اول با استفاده از الگوریتم پیشنهادی**

معادله‌ی مرتبه‌ی اول برای یک مسئله‌ی میدان‌گذرا پس از منفصل‌سازی در مکان به فرم معادلات درگیر می‌باشند که می‌توان آنها را با استفاده از تکنیک جداسازی مودال به معادلات غیردرگیر تبدیل نمود، بنابراین الگوریتم پیشنهادی را بر روی یک معادله‌ی مرتبه‌ی اول یک سیستم یک درجه آزادی مطابق زیر اعمال می‌کنیم:

$$\dot{y}(t)+\lambda y(t)=f(t) \tag{17}$$

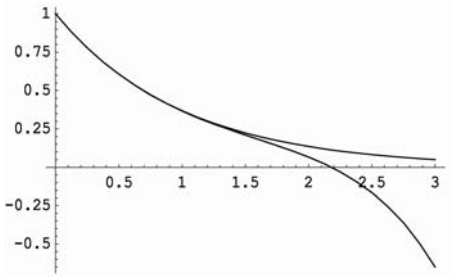
جواب دقیق معادله‌ی فوق عبارت است از:

$$\bar{y}(t)=y_0\,e^{-\lambda t}+\int_0^t e^{-\lambda(t-\tau)}f(\tau)\,d\tau \tag{18}$$

که جمله‌ی اول سمت راست رابطه‌ی فوق، جواب هموژن معادله را نشان می‌دهد. بسط تیلور تابع $e^{-x}$ مطابق رابطه‌ی زیر می‌باشد:

$$e^{-x}=\sum_{k=0}^{\infty}\frac{(-x)^k}{k!} \tag{19}$$

با انتخاب جملات محدود از بسط فوق دامنه‌ی انطباق بسط به تابع محدودتر می‌شود. مثلاً در حالت استفاده از شش جمله از بسط فقط تا حدود $x=1.0$ رابطه‌ی (۱۹) دقت کافی دارد، به شکل (۱) توجه شود.




**شکل ۱ – تقریب تابع نمایی با بسط به سری تیلور آن**

از آنجا که هدف ما تقریب جواب در فاصله‌ی گام زمانی $\Delta t$ است لذا با انتخاب تعداد مناسب از جملات بسط تیلور تابع جواب دقیق، می‌توان جواب دقیق را با بسط تیلور آن در فاصله‌ی گام زمانی با جملات محدود ( و مرتبط با فاصله گام زمانی ) تقریب زد. با انتخاب $\hat{t}=t/\Delta t$ , $0\le\hat{t}\le 1$ رابطه‌ی (۱۷) به فرم زیر در می‌آید:

$$\dot{y}(\hat{t})+\lambda\,\Delta t\,y(\hat{t})=\Delta t\,f(\hat{t}) \tag{20}$$

که جواب هموژن معادله‌ی فوق با فرض $y_0=1$ به صورت $e^{-\lambda\,\Delta t\,\hat{t}}$ است بنابراین در انتهای گام زمانی خواهیم داشت:

$$\bar{y}(\Delta t)=\bar{y}(1)=e^{-\lambda\,\Delta t} \tag{21}$$

این نکته که محدوده‌ی انطباق یک تابع با بسط تیلور آن، بستگی به تعداد جملات بسط تیلور تابع دارد تقریباً در مورد تمامی توابعی که بسط تیلور دارند صادق است. ازآنجا که در الگوریتم پیشنهادی انتگرال‌گیری گام به گام، از بسط تیلور جواب دقیق برای باز تولید آن استفاده می‌کنیم، بنابراین با توجه به گام زمانی موردنظر تعداد جملات بسط تیلور انتخاب می‌شود.



Computer For Civil Software Engineering Group : www.CCSofts.com , www.CompCivil.com

با توجه به بحث‌های بخش قبل تابع تقریبی را در فاصله‌ی $0 \le \hat{t} \le 1$ به صورت زیر در نظر می‌گیریم:

$$y^a(\hat{t}) = \sum_{j=1}^{NP} N_j(\hat{t})\,\hat{y}_j \qquad (22)$$

حال با تقریب جواب دقیق با تابع فوق، باید معادلات حاکم و شرایط مرزی را در نقاط همجایگی(NP نقطه) ارضاء کنیم؛ اگر Nd تعداد نقاطی باشد که شرط مرزی در آن نقاط داریم (در مورد مسئله‌ی ما فقط یک شرط مرزی در $\hat{t} = 0$ داریم یعنی Nd = 1 می‌باشد)، در باقی نقاط همجایگی یعنی Nr (Np-Nd) = نقطه نیز معادله‌ی حاکم را باید ارضاء کنیم، بنابراین:

$$\begin{aligned} & y^a(\hat{t}_1) = y_0 \\ & L(y^a(\hat{t}_i)) = \Delta t\; f(\hat{t}_i) \quad ; \quad i = 2,3,\cdots,NP \end{aligned} \qquad (23)$$

از معادلات (۲۳) می‌توان به معادله‌ی زیر رسید و با حل آن $\hat{\mathbf{y}}$ را بدست آورد:

$$\widetilde{\mathbf{K}} \cdot \hat{\mathbf{y}} = \mathbf{b} \quad ; \quad \widetilde{\mathbf{K}} \in R^{NP\times NP} \quad ; \quad \hat{\mathbf{y}} \in R^{NP\times 1} \quad ; \quad \mathbf{b} \in R^{NP\times 1} \qquad (24)$$

ماتریس $\widetilde{\mathbf{K}}$ با جایگذاری رابطه‌ی (۲۲) در روابط (۲۳)، از روابط زیر محاسبه می‌شود:

$$\begin{aligned} & \widetilde{\mathbf{K}}_{1j} = N_j(\hat{t}_1) \quad ; \quad j = 1,2,\cdots,NP \\ & \widetilde{\mathbf{K}}_{ij} = \frac{\partial N_j(\hat{t}_i)}{\partial \hat{t}} + \lambda \Delta t\, N_j(\hat{t}_i) \quad ; \quad \begin{cases} i = 2,3,\cdots,NP \\ j = 1,2,\cdots,NP \end{cases} \end{aligned} \qquad (25)$$

در بردار $\mathbf{b}$ اولین درایه شرط مرزی $y_0$ است. در مورد درایه‌های بعدی چون نیروی محرک f فقط در اول گام و انتهای گام زمانی مورد نظر یعنی $(\hat{t} = 1\,,\,\hat{t} = 0)$ معین است بایستی در نقاط بین گام زمانی به نحوی تقریب زده شود. یکی از نقاط قوت الگوریتم پیشنهادی این است که از تقریب‌های دلخواه برای بارگذاری در این فاصله‌ی گام زمانی می‌توان استفاده کرد و یا حتی در حالت خیلی مطلوب می‌توان مقادیر دقیق بارگذاری را در بین گام زمانی به صورت ورودی الگوریتم دریافت کرد بدون اینکه فاصله‌های گام زمانی حل را کوچکتر کنیم.

پس از حل معادله‌ی (۲۴)، بردار $\hat{\mathbf{y}}$ مشخص می‌شود و می‌توان جواب را از رابطه‌ی زیر در هر نقطه‌ی دلخواه گام زمانی و همچنین در انتهای گام زمانی تعیین نمود:

$$y^a(\hat{t}_i) = \sum_{j=1}^{NP} N_j(\hat{t}_i)\,\hat{y}_j \quad ; \quad y^a(1) = \sum_{j=1}^{NP} N_j(1)\,\hat{y}_j \qquad (26)$$

بنابر رابطه‌ی فوق با داشتن مقادیر $y_0$ , $f_0$ , $f_1$ در هر گام زمانی $y_1$ در انتهای گام زمانی بدست می‌آید در واقع رابطه‌ی (۲۶) رابطه‌ی بازگشتی الگوریتم پیشنهادی است و به طور شماتیک می‌توان نوشت:

$$y_{n+1} = g(y_n, f_n, f_{n+1}) \qquad (27)$$

مراحل الگوریتم پیشنهادی را می‌توان در جدول (۱) خلاصه نمود.

**جدول ۱- مراحل الگوریتم پیشنهادی**

| |
|---|
| I. محاسبات اولیه ( فقط یک بار ) |
| انتخاب تابع ویندو |
| انتخاب تعداد جملات برای تقریب بسط تیلور با توجه به گام زمانی مطلوب |
| ارضاء شرایط بازیابی و بدست آوردن ضرایب تابع تصحیح |
| تشکیل معادله $\widetilde{\mathbf{K}} \cdot \hat{\mathbf{y}} = \mathbf{b}$ و حل آن |
| تشکیل رابطه‌ی بازگشتی به فرم $y^a(1) = \sum_{j=1}^{NP} N_j(1)\,\hat{y}_j \;\rightarrow\; y_{n+1} = g(y_n, f_n, f_{n+1})$ |
| II. برای هرگام زمانی |
| مشخص نمودن $f_n$ , $f_{n+1}$ (مقادیر بار در شروع و انتهای گام زمانی ) |
| محاسبه $y_{n+1}$ از رابطه‌ی بازگشتی فوق برای هر گام زمانی |



Computer For Civil Software Engineering Group : www.CCSofts.com , www.CompCivil.com

تابع ویندو را می‌توان تابع گاوسین یا اسپیلاین انتخاب نمود؛ تابع اسپیلاین مکعب به صورت زیر می‌باشد:

$$\phi(t)=\begin{cases}(t+2)^3/6 & -2\le t\le -1\\ 2/3-t^2(1+t/2) & -1\le t\le 0\\ 2/3-t^2(1-t/2) & 0\le t\le 1\\ -(t-2)^3/6 & 1\le t\le 2\end{cases} \tag{28}$$

**تعمیم الگوریتم برای سیستمهای چند درجه آزادی**

معادله‌ی حاکم بر یک مساله‌ی میدان گذرا پس از منفصل سازی مکانی به فرم معادله زیر می‌باشد:

$$\mathbf{C}\,\dot{\mathbf{y}}(t)+\mathbf{K}\,\mathbf{y}(t)=\mathbf{f}(t) \tag{29}$$

با شرط اولیه‌ی $\mathbf{y}(0)=\{\mathbf{y}_0\}$ که در آن $\mathbf{y}(t)=[y^1(t)\ y^2(t)\cdots y^n(t)]^T$ و $\mathbf{f}(t)=[f^1(t)\ f^2(t)\cdots f^n(t)]^T$ و n تعداد درجات آزادی سیستم می باشد.

هدف ما تقریب جوابهای درجات آزادی در فاصله‌ی گام زمانی $\Delta t$ است. با انتخاب $\hat{t}=t/\Delta t$ , $0\le\hat{t}\le 1$ رابطه‌ی (۲۹) به فرم زیر در می‌آید.

$$\mathbf{C}\,\dot{\mathbf{y}}(\hat{t})+\Delta t\,\mathbf{K}\,\mathbf{y}(\hat{t})=\Delta t\,\mathbf{f}(\hat{t}) \tag{30}$$

با توجه به بحثهای بخش قبلی توابع تقریبی را در فاصله‌ی $0\le\hat{t}\le 1$ به صورت زیر در نظر می‌گیریم .

$${}^a y^i(\hat{t})=\sum_{j=1}^{NP}N_j(\hat{t})\hat{y}_j^i \tag{31}$$

حال بایستی شرط اولیه و معادله‌ی حاکم را در نقاط همجایگی ارضاء کنیم که در نهایت به دستگاه معادلاتی می‌رسیم که $\hat{y}_j^i$ ها از حل این معادلات تعیین می‌شوند. در $\hat{t}=\hat{t}_1=0$ باید شرط اولیه ارضاء گردد ودر باقی نقاط همجایگی باید معادله‌ی حاکم ارضاء شود یعنی:

$$\begin{aligned}&{}^a y^i(\hat{t}_1)=y_0^i\\ &L({}^a y(\hat{t}_k))=\Delta t\ f(\hat{t}_k)\quad;\quad k=2,3,\cdots,NP\end{aligned} \tag{32}$$

در نهایت از معادلات (۳۲) به معادله‌ی زیر می‌رسیم که با حل آن $\hat{y}_j^i$ ها بدست می‌آیند.

$$\widetilde{\mathbf{K}}\,\hat{\mathbf{y}}=\mathbf{b}\quad;\quad \widetilde{\mathbf{K}}\in R^{(NP\times n)\times(NP\times n)}\quad;\quad \hat{\mathbf{y}}\in R^{(NP\times n)\times 1}\quad;\quad \mathbf{b}\in R^{(NP\times n)\times 1} \tag{33}$$

ماتریس $\widetilde{\mathbf{K}}$ و بردارهای $\hat{\mathbf{y}},\mathbf{b}$ با جایگذاری رابطه‌ی (۳۱) در روابط (۳۲)، مطابق رابطه‌ی (۳۴) محاسبه می‌شوند:

$$\begin{bmatrix}\overbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}^{n}\\ NP\left\{\begin{matrix}n\left\{\begin{bmatrix}\overbrace{N_j(\hat{t}_1)}^{NP}\\ \vdots\\ 0\end{bmatrix}\right. & n\left\{\begin{bmatrix}\overbrace{\cdots}^{NP}\\ \ddots\\ \cdots\end{bmatrix}\right. & n\left\{\begin{bmatrix}\overbrace{0}^{NP}\\ \vdots\\ N_j(\hat{t}_1)\end{bmatrix}\right.\\ \begin{bmatrix}c_{11}\dot{N}_j(\hat{t}_2)+\Delta t\,k_{11}N_j(\hat{t}_2)\\ \vdots\\ c_{n1}\dot{N}_j(\hat{t}_2)+\Delta t\,k_{n1}N_j(\hat{t}_2)\end{bmatrix} & \begin{bmatrix}\cdots\\ \ddots\\ \cdots\end{bmatrix} & \begin{bmatrix}c_{1n}\dot{N}_j(\hat{t}_2)+\Delta t\,k_{1n}N_j(\hat{t}_2)\\ \vdots\\ c_{nn}\dot{N}_j(\hat{t}_2)+\Delta t\,k_{nn}N_j(\hat{t}_2)\end{bmatrix}\\ \vdots & \vdots & \vdots\\ \begin{bmatrix}c_{11}\dot{N}_j(\hat{t}_{NP})+\Delta t\,k_{11}N_j(\hat{t}_{NP})\\ \vdots\\ c_{n1}\dot{N}_j(\hat{t}_{NP})+\Delta t\,k_{n1}N_j(\hat{t}_{NP})\end{bmatrix} & \begin{bmatrix}\cdots\\ \ddots\\ \cdots\end{bmatrix} & \begin{bmatrix}c_{1n}\dot{N}_j(\hat{t}_{NP})+\Delta t\,k_{1n}N_j(\hat{t}_{NP})\\ \vdots\\ c_{nn}\dot{N}_j(\hat{t}_{NP})+\Delta t\,k_{nn}N_j(\hat{t}_{NP})\end{bmatrix}\end{matrix}\right.\end{bmatrix}\bullet\begin{bmatrix}n\left\{\begin{matrix}NP\left\{\begin{bmatrix}\hat{y}_1^1\\ \vdots\\ \hat{y}_{NP}^1\end{bmatrix}\right.\\ NP\left\{\begin{bmatrix}\hat{y}_1^2\\ \vdots\\ \hat{y}_{NP}^2\end{bmatrix}\right.\\ \vdots\\ NP\left\{\begin{bmatrix}\hat{y}_1^n\\ \vdots\\ \hat{y}_{NP}^n\end{bmatrix}\right.\end{matrix}\right.\end{bmatrix}=\begin{bmatrix}NP\left\{\begin{matrix}n\left\{\begin{bmatrix}y_0^1\\ \vdots\\ y_0^n\end{bmatrix}\right.\\ n\left\{\begin{bmatrix}\Delta t\ f_2^1\\ \vdots\\ \Delta t\ f_2^n\end{bmatrix}\right.\\ \vdots\\ n\left\{\begin{bmatrix}\Delta t\ f_{NP}^1\\ \vdots\\ \Delta t\ f_{NP}^n\end{bmatrix}\right.\end{matrix}\right.\end{bmatrix} \tag{34}$$

پس از تعیین بردار $\hat{\mathbf{y}}$ ، می توان جواب را در هر نقطه‌ی دلخواه گام زمانی و همچنین در انتهای گام زمانی (رابطه‌ی بازگشتی الگوریتم)، تعیین نمود.

$${}^a y^i(\hat{t}_k)=\sum_{j=1}^{NP}N_j(\hat{t}_k)\,\hat{y}_j^i\quad\Rightarrow\quad {}^a y^i(1)=\sum_{j=1}^{NP}N_j(1)\,\hat{y}_j^i \tag{35}$$



Computer For Civil Software Engineering Group : www.CCSofts.com , www.CompCivil.com

**مثال‌های عددی**

**م ۱)** مطلوبست حل معادله‌ی $\dot{y}+\lambda y=f(t)$ به ازاء $\lambda=10\,,100$ و $f(t)=1000\left(1-\frac{t}{0.5}\right)$ (شکل (۲) ) و شرط اولیه‌ی $y(0)=0$.

الف ) حل تحلیلی

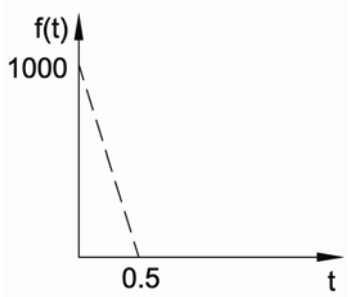
$$\bar{y}(t)=1000\int_0^t e^{-\lambda(t-\tau)}(1-\frac{\tau}{0.5})\,d\tau=1000\left(\frac{2(-1-0.5\lambda)e^{-\lambda t}}{\lambda^2}-\frac{2(-1-0.5\lambda-\lambda t)}{\lambda^2}\right)\qquad 0\le t\le 0.5$$

$$\bar{y}(t)=1000\int_0^{0.5} e^{-\lambda(t-\tau)}(1-\frac{\tau}{0.5})\,d\tau=1000\left(\frac{2(-1-0.5\lambda)e^{-\lambda t}}{\lambda^2}-\frac{2e^{0.5\lambda-\lambda t}}{\lambda^2}\right)\qquad 0.5\le t$$

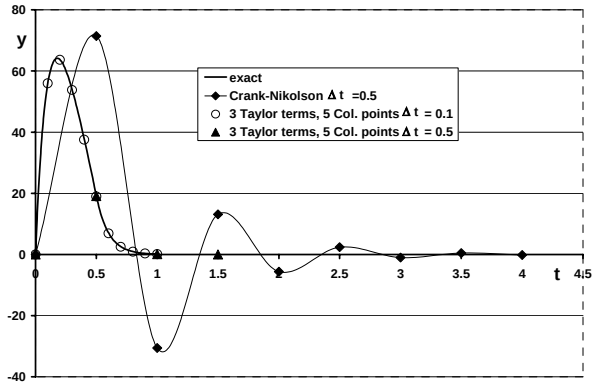
ب ) حل بروش $\theta$ ( روش کرنک- نیکلسون $\theta=1/2$ )؛ رابطه بازگشتی روش کرنک- نیکلسون به صورت زیر خلاصه می‌شود:

$$y_{n+1}=(1+\frac{\Delta t}{2}\lambda)^{-1}[(1-\frac{\Delta t}{2}\lambda)y_n+\bar{f}]\quad;\quad \bar{f}=\frac{1}{2}\Delta t(f_{n+1}+f_n)$$

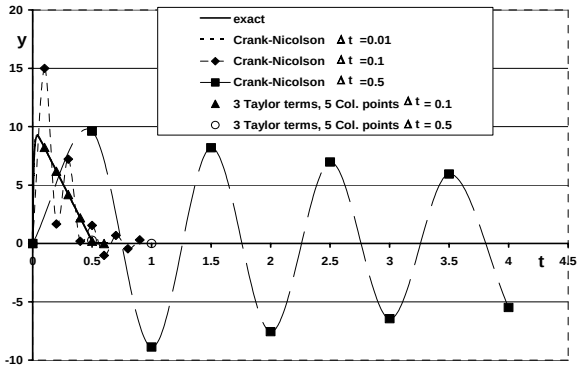
ج ) حل بروش الگوریتم پیشنهادی؛ از رابطه بازگشتی (۲۷) استفاده می‌کنیم. برای $\lambda=10$ و $\lambda=100$ نتایج به ازای $\Delta t=0.5,\Delta t=0.1$ در اشکال (۳) و (۴) آورده شده است. یکی از مزایای مهم الگوریتم پیشنهادی این است که در میان گام زمانی نیز می‌توان جواب دقیق را برآورد کرد. برای مثال فوق در حالتی که $\lambda=100$ و $\Delta t=0.5$ است، در نقاطی از بین گام زمانی نیز جواب برآورد شده که در شکل (۵) نشان داده شده‌اند.



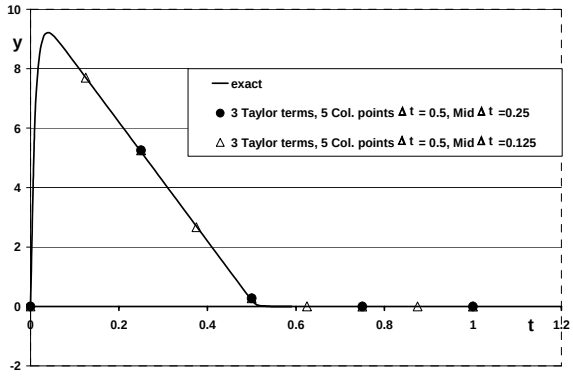

**شکل ۲ – تابع ضربه**




**شکل ۳ – نتایج تحلیل معادله‌ی مرتبه‌ی اول برای** $\lambda=10$




**شکل ۴ – نتایج تحلیل معادله‌ی مرتبه‌ی اول برای** $\lambda=100$




**شکل ۵ – نتایج تحلیل برای** $\lambda=100$ **با برآورد جواب در نقاط بین گام زمانی**

**م ۲)** مطلوب است حل معادله‌ی $\begin{bmatrix}1&-1\\-1&2\end{bmatrix}\begin{Bmatrix}\dot{y}^1(t)\\\dot{y}^2(t)\end{Bmatrix}+\begin{bmatrix}4&2\\2&6\end{bmatrix}\begin{Bmatrix}y^1(t)\\y^2(t)\end{Bmatrix}=\begin{Bmatrix}f^1(t)\\f^2(t)\end{Bmatrix}$ با شرط اولیه‌ی $\begin{Bmatrix}y^1(0)\\y^2(0)\end{Bmatrix}=\begin{Bmatrix}0\\0\end{Bmatrix}$ و بردار بار

$$\begin{cases}f^1(t)=500(1-\frac{t}{0.1})\quad;\quad 0\le t\le 0.1\\ f^2(t)=1000(1-\frac{t}{0.1})\quad;\quad 0\le t\le 0.1\end{cases}.$$



Computer For Civil Software Engineering Group : www.CCSofts.com , www.CompCivil.com

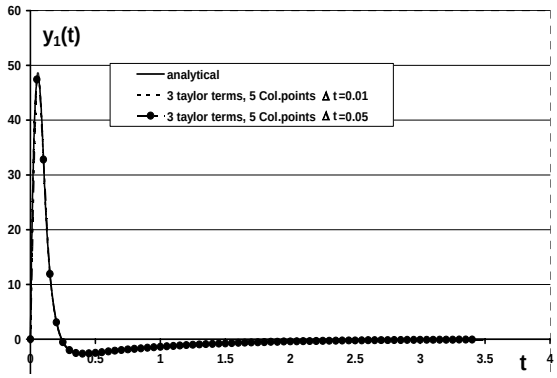
الف ) حل تحلیلی؛ با استفاده از جدا سازی مودال می‌توان معادلات را به صورت مستقل بیان نمود و در نهایت حل تحلیلی معادلات به فرم زیر می‌باشد:

$$\bar{y}^1(t) = 0.4814 * \bar{y}_l^1(t) + 1.3298 * \bar{y}_l^2(t) \qquad ; \qquad \bar{y}^2(t) = -0.42415 * \bar{y}_l^1(t) + 0.90557 * \bar{y}_l^2(t)$$

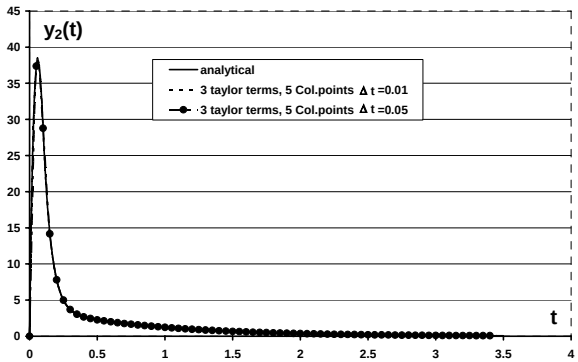
$$\begin{cases} 0 \le t \le 0.1 \rightarrow \bar{y}_l^1(t) = -183.45(-7.9045\, e^{-1.1898\, t} - 8.40477(-0.940477 + t)) \\ t \ge 0.1 \qquad \rightarrow \bar{y}_l^1(t) = -183.45(7.06402\, e^{0.11898 - 1.1898\, t} - 7.9045\, e^{-1.1898\, t}) \end{cases}$$

$$\begin{cases} 0 \le t \le 0.1 \rightarrow \bar{y}_l^2(t) = 1570.67(-0.09488\, e^{-16.8102\, t} - 0.59488(-0.1595 + t)) \\ t \ge 0.1 \qquad \rightarrow \bar{y}_l^2(t) = 1570.67(0.03539\, e^{1.68102 - 16.8102\, t} - 0.09488\, e^{-16.8102\, t}) \end{cases}$$

ب ) حل به روش الگوریتم پیشنهادی؛ از الگوریتم با سه جمله‌ی بسط به سری تیلور تابع جواب و پنج نقطه‌ی همجایگی استفاده می‌کنیم، نتایج تحلیل به ازاء گام‌های زمانی 0.01sec , 0.05 در اشکال (۶) و (۷) نشان داده شده‌اند.




شکل ۶ – نتایج تحلیل سیستم دو درجه آزادی (درجه‌ی آزادی ۱ )




شکل ۷ – نتایج تحلیل سیستم دو درجه آزادی (درجه‌ی آزادی ۲ )

## نتیجه گیری

در این تحقیق با به‌کاربردن مفهوم باز تولید توابع یک الگوریتم گام‌به‌گام نوینی برای حل معادلات مرتبه‌ی اول ارائه شده است. با استفاده از این الگوریتم می‌توان گام‌های زمانی بزرگ را در حل به کار برد. این امر مستلزم انتخاب مناسب تعداد جملات بسط به سری تیلور تابع جواب و همچنین تعداد نقاط همجایگی مناسب برای ارضای معادلات حاکم می‌باشد، تا بتوان با این گام‌های زمانی بزرگ جواب دقیق را باز تولید نمود. در این الگوریتم استخراج روابط بازگشتی الگوریتم مرتبط با گام زمانی موردنظر می‌باشد بنابراین تا فاصله‌ی گام زمانی‌ایی که روابط بازگشتی الگوریتم نسبت به آن استخراج شده‌اند انتظار داریم نتایج بدست‌آمده دقیق باشند. ضمناً در شرایطی که گام‌های زمانی حل بزرگ هستند می‌توان جواب را در نقاط بین گامی نیز برآورد کرد بدون اینکه نیازی به ریز کردن گامهای زمانی حل باشد.

## مراجع


۱- سلیمانی شیشوان، س. (۱۳۸۳) حل مسائل تئوری ارتعاشات توسط اَبَرگامهای زمانی با ابداع جدید در روش انتگرال‌گیری گام‌به‌گام. پایان‌نامه کارشناسی ارشد، دانشکده‌ی عمران ـ پردیس دانشکده‌های فنی ـ دانشگاه تهران

2. Zienkiewicz O.C, Talor R.L. (1991) *The finite element method*. 4th Ed. Vol. 2 McGRAW-HILL International.
3. Fung T.C. (1999) Complex-time-step methods for transient analysis. *Int. J. Numr. Meth. Engng*; **46**, 1253-71
4. Fung T.C. (1998) Higher order time-step integration methods with complex time steps. *Journal of Sound and Vibration*; **210**, 69-89.
5. Fung T.C. (1999) Weighting parameters for unconditionally stable higher-order accurate time step integration algorithms.PART1- First-order equations. *Int. J. Numr. Meth. Engng*; **45**, 941-970.
6. Fung T.C. (2000) Unconditionally stable higher-order accurate collocation time-step algorithms for first-order equations. *Comput. Methods Appl. Mech. Engrg*; **190**, 1651-1662.
7. Fung T.C. (1999) Higher-order accurate least-squares methods for first-order initial value problems. *Int. J. Numr. Meth. Engng*; **45**, 77-99
8. Liu W.K, Chen Y, Uras R.A, Chang C.T. (1996) Generalized multiple scale reproducing kernel particle methods *Comput. Methods Appl. Mech. Engrg*6; **139**, 91-157.
9. Aluru N.R. (2000) A point collocation methods based on reproducing kernel approximations. *Int. J. Numr. Meth. Engng*; **47**, 1083-1121.




Computer For Civil Software Engineering Group : www.CCSofts.com , www.CompCivil.com