# ON MATCHING THE SSpj AND GNpj ALGORITHMS

by

W.L. WOOD

---

NUMERICAL ANALYSIS REPORT 19/85

Department of Mathematics

University of Reading

Submitted to "Communications in Applied Numerical Methods"

The SSpj algorithm is a single-step method for the integration of

$$M\ddot{\underline{u}} + C\dot{\underline{u}} + K\underline{u} = \underline{f} \ , \quad (j=2) \tag{1}$$

or

$$C\dot{\underline{u}} + K\underline{u} = \underline{f} \ , \quad (j=1) \tag{2}$$

using a pth degree approximation for the unknown $u(t)$ in a time step and the satisfaction of a Weighted Residual equation. This algorithm has $p$ parameters $\theta_1,\ldots,\theta_p$. [1] We take $p \geq j$.

The GNpj is a generalized Newmark or 'β-p' method introduced in reference [2] for equation (1) but equally applicable to equation (2). This also uses $p$ parameters and approximations constructed in the Newmark manner and with collocation at time $t = (n+1)\Delta t$.

For the purposes of the discussion in this paper it is sufficient to consider the SSpj and GNpj algorithms applied to the scalar equation which we assume can be obtained from the modal decomposition of the homogeneous form of equation (1):

$$m\ddot{u} + c\dot{u} + ku = 0 \quad . \tag{3}$$

The SSpj and GNpj algorithms do not appear in the form where we can make a straightforward comparison of their amplification matrices. Looking first at the SSpj algorithm and referring for details to reference [1], we have there equations (6) for $u_{n+1}, \dot{u}_{n+1},\ldots,{}^{p-1}u_{n+1}$. When the substitution for $\alpha_n^{(p)}$ from equation (11) in reference [1] is made we have the SSpj algorithm in the form

$$\underline{X}_{n+1}^{(s)} = A^{(s)}\underline{X}_n^{(s)} \tag{4}$$

where $\underline{X}_n^{(s)} = [u_n, \Delta t\dot{u}_n,\ldots,\Delta t^{p-1}\,{}^{p-1}u_n]^T$ and the amplification matrix $A^{(s)}$ is of order $p$.

The equations for the GNpj (or β-p) algorithm can be found in reference [2]. For convenience in matching the SSpj and GNpj algorithms we are here replacing $\beta_0,\beta_1,\ldots,\beta_{p-1},\beta_p$ by $\gamma_p,\gamma_{p-1},\ldots,\gamma_1,\gamma_0$ respectively with

$\gamma_0 = 1$ corresponding to $\beta_p = 1$. Using reference [2] equations (6) which define $u_{n+1}, \dot{u}_{n+1}, \ldots, \overset{p}{u}_{n+1}$ in terms of $u_n, \dot{u}_n, \ldots, \overset{p}{u}_n$ and

$$\Delta \overset{p}{u}_n = \overset{p}{u}_{n+1} - \overset{p}{u}_n \tag{5}$$

we see that substituting for $\Delta \overset{p}{u}$ from reference [2] equation (8) gives a system of equations

$$\underline{X}^{(G)}_{n+1} = A^{(G)} \underline{X}^{(G)}_n \tag{6}$$

where $\underline{X}^{(G)}_n = [u_n, \Delta t \dot{u}_n, \ldots, \Delta t^p \overset{p}{u}_n]^T$ and the GNpj amplification matrix $A^{(G)}$ is of order p+1. (The GNpj algorithm applied to the system of equations (1) has the disadvantage over SSpj of having to carry an additional vector $\underline{\overset{p}{u}}_n$ from one time step to the next).

Hence it is more convenient to make an indirect approach to the matching of the two algorithms. Considering first the SSp2 algorithm applied to equation (3) we write first the equation corresponding to reference [1] equation (11) and then the equations corresponding to reference [1] equations (6) with $u_{n+1} = \lambda u_n, \dot{u}_{n+1} = \lambda \dot{u}_n, \ldots\ldots, \overset{p-1}{u}_{n+1} = \lambda \overset{p-1}{u}_n$ so that we have a system of equations

$$S \underline{y}_n = \underline{0} \tag{7}$$

where $\underline{y}_n^T = [u_n, \Delta t \dot{u}_n, \Delta t^2 \ddot{u}_n, \ldots, \Delta t^{p-1} \overset{p-1}{u}_n, \Delta t^p \alpha_n^p]$ and S is the (p+1) order matrix:

$$S = \begin{pmatrix} b_0 & b_1 & b_2 & \cdot & \cdot & \cdot & \cdot b_{p-1} & b_p \\ 1-\lambda & 1 & \frac{1}{2} & \cdot & \cdot & \cdot & \frac{1}{(p-1)!} & \frac{1}{p!} \\ 0 & 1-\lambda & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & & & & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & 1-\lambda & 1 & \frac{1}{2!} \\ 0 & 0 & \cdot & \cdot & \cdot & 0 & 1-\lambda & 1 \end{pmatrix} \tag{8}$$

where $b_0 = \theta_0 k\Delta t^2$, $b_1 = \theta_0 c\Delta t + \theta_1 k\Delta t^2$,

$$b_q = \frac{m}{(q-2)!}\theta_{q-2} + \frac{c\Delta t}{(q-1)!}\theta_{q-1} + \frac{k\Delta t^2}{q!}\theta_q, \quad q = 2,3,\ldots,p \tag{9}$$

($\theta_0 \equiv 1$).

Now we take the GNp2 algorithm applied to equation (2) as given by reference [2] equations (6) with the $\Delta u^{(p)}$ terms substituted for from reference [2] equation (8) and put $u_{n+1} = \lambda u_n$, $\dot{u}_{n+1} = \lambda \dot{u}_n, \ldots, {}^{p}u_{n+1} = \lambda {}^{p}u_n$. The result is the system of $p+1$ homogeneous equations

$$G\underline{X}_n^{(G)} = \underline{0} \tag{10}$$

where $\underline{X}_n^{(G)}$ is as in equation (6) and $G$ is the matrix given by

$$G = \begin{bmatrix} a_0 & a_1 & a_2 & \cdot & \cdot & a_{p-1} & a_p \\ 1-\lambda & 1 & \tfrac{1}{2}! & \cdot & \cdot & \frac{1}{(p-1)!} & \frac{1}{p!}[1+\gamma_p(\lambda-1)] \\ 0 & 1-\lambda & 1 & \cdot & \cdot & \frac{1}{(p-2)!} & \frac{1}{(p-1)!}[1+\gamma_{p-1}(\lambda-1)] \\ \cdot & \cdot & \cdot & & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & & \cdot & \cdot \\ 0 & 0 & \cdot & 1-\lambda & 1 & & \tfrac{1}{2}![1+\gamma_2(\lambda-1)] \\ 0 & 0 & \cdot & 0 & 1-\lambda & & 1+\gamma_1(\lambda-1) \end{bmatrix} \tag{11}$$

where $a_0 = k\Delta t^2$, $a_1 = c\Delta t + k\Delta t^2$

$$a_q = \frac{m}{(q-2)!} + \frac{c\Delta t}{(q-1)!} + \frac{k\Delta t^2}{q!}, \quad q = 2,3,\ldots,(p-1)$$

and $$a_p = \frac{m}{(p-2)!}[1+\gamma_{p-2}(\lambda-1)] + \frac{c\Delta t}{(p-1)!}[1+\gamma_{p-1}(\lambda-1)] + \frac{k\Delta t^2}{p!}[1+\gamma_p(\lambda-1)]. \tag{12}$$

Then since the system of equations (7) only gives a non-trivial solution for $\underline{Y}_n$ if determinant (S) = 0, this equation in $\lambda$ is the stability polynomial for the SSp2 algorithm. Similarly determinant (G) = 0 gives the

stability polynomial for the GNp2 algorithm.

Suppose

$$\det(S) \equiv mS_1 + c\Delta tS_2 + k\Delta t^2 S_3 \tag{13}$$

Then from equation (8)

$$S_3 = \begin{pmatrix} 1 & \theta_1 & \frac{\theta_2}{2!} & \cdot & \cdot & \cdot & \frac{\theta_{p-1}}{(p-1)!} & \frac{\theta_p}{p!} \\ 1-\lambda & 1 & \frac{1}{2!} & \cdot & \cdot & \cdot & \frac{1}{(p-1)!} & \frac{1}{p!} \\ 0 & 1-\lambda & 1 & \cdot & \cdot & \cdot & \cdot \quad \cdot & \cdot \quad \cdot \\ \cdot & \cdot & \cdot & & & & \cdot & \cdot \quad \cdot \\ 0 & 0 & \cdot & \cdot & \cdot & 1-\lambda & 1 & \frac{1}{2!} \\ 0 & 0 & \cdot & \cdot & \cdot & 0 & 1-\lambda & 1 \end{pmatrix} \tag{14}$$

The value of the determinant is unaltered by adding to row 2 the result of $(\lambda-1)$ times row 1 which gives $S_3 = \Delta_p$ where $\Delta_p$ is the pth order determinant

$$\Delta_p = \begin{pmatrix} 1+\theta_1(\lambda-1) & \frac{1}{2!}[1+\theta_2(\lambda-1)] & \cdot & \frac{1}{(p-1)!}[1+\theta_{p-1}(\lambda-1)] & \frac{1}{p!}[1+\theta_p(\lambda-1)] \\ 1-\lambda & 1 & \cdot \quad \cdot & \frac{1}{(p-2)!} & \frac{1}{(p-1)!} \\ 0 & 1-\lambda & \cdot \quad \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \quad \cdot & \cdot & \cdot \\ 0 & 0 & \cdot \quad 1-\lambda & 1 & \frac{1}{2!} \\ 0 & 0 & \cdot \quad 0 & 1-\lambda & 1 \end{pmatrix} \tag{15}$$

Now we take

$$\det(G) \equiv G_1 m + G_2 c\Delta t + G_3 k\Delta t^2 \tag{16}$$

Then we have

$$G_3 = \begin{pmatrix} 1 & 1 & \frac{1}{2!} & \cdot & \cdot & \frac{1}{(p-1)!} & \frac{1}{p!}[1 + \gamma_p(\lambda-1)] \\ 1-\lambda & 1 & \frac{1}{2!} & \cdot & \cdot & \frac{1}{(p-1)!} & \frac{1}{p!}[1 + \gamma_p(\lambda-1)] \\ 0 & 1-\lambda & 1 & \cdot & \cdot & \frac{1}{(p-2)!} & \frac{1}{(p-1)!}[1 + \gamma_{p-1}(\lambda-1)] \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & 1-\lambda & 1 & & \frac{1}{2!}[1 + \gamma_2(\lambda-1)] \\ 0 & 0 & \cdot & 0 & 1-\lambda & & 1 + \gamma_1(\lambda-1) \end{pmatrix} \tag{17}$$

Subtract row 2 from row 1 then we have

$$G_3 = \lambda \begin{pmatrix} 1 & \frac{1}{2!} & \cdot & \cdot & \frac{1}{(p-1)!} & \frac{1}{p!}[1 + \gamma_p(\lambda-1)] \\ 1-\lambda & 1 & \cdot & \cdot & \frac{1}{(p-2)!} & \frac{1}{(p-1)!}[1 + \gamma_{p-1}(\lambda-1)] \\ 0 & 1-\lambda & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & 1-\lambda & 1 & \frac{1}{2!}[1 + \gamma_2(\lambda-1)] \\ 0 & 0 & \cdot & 0 & 1-\lambda & 1 + \gamma_1(\lambda-1) \end{pmatrix} \tag{18}$$

$$= \lambda\Delta_p \quad \text{if} \quad \theta_j = \gamma_j, \quad j = 1,2,\ldots,p \tag{19}$$

We next consider

$$S_2 = \begin{pmatrix} 0 & 1 & \theta_1 & \cdot & \cdot & \frac{\theta_{p-1}}{(p-1)!} \\ 1-\lambda & 1 & \frac{1}{2!} & \cdot & \cdot & \frac{1}{p!} \\ 0 & 1-\lambda & 1 & \cdot & \cdot & \frac{1}{(p-1)!} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & 1-\lambda & 1 & \frac{1}{2!} \\ 0 & 0 & \cdot & 0 & 1-\lambda & 1 \end{pmatrix} \tag{20}$$

$$= (\lambda-1)\begin{pmatrix} 1 & \theta_1 & \frac{\theta_2}{2!} & \cdot & \cdot & \frac{\theta_{p-1}}{(p-1)!} \\ 1-\lambda & 1 & \frac{1}{2!} & \cdot & \cdot & \frac{1}{(p-1)!} \\ 0 & 1-\lambda & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & 1-\lambda & 1 & \frac{1}{2!} \\ 0 & 0 & \cdot & 0 & 1-\lambda & 1 \end{pmatrix}$$

Then repeating the maneouvre row 2 + $(\lambda-1)$ times row 1 which was used on $S_3$ gives:

$$S_2 = (\lambda-1)\Delta_{p-1} \tag{21}$$

Manipulating $G_2$ in a similar way to that used on $G_3$ then gives

$$G_2 = \lambda(\lambda-1)\Delta_{p-1} \quad \text{if} \quad \theta_j = \gamma_j, \quad j = 1,2,\ldots,p \tag{22}$$

Similarly $S_1 = (\lambda-1)^2\Delta_{p-2}, \quad (\Delta_0 = 1)$ (23)

and $G_1 = \lambda S_1$ again if $\theta_j = \gamma_j, \quad j = 1,2,\ldots,p$ . (24)

Thus, apart from the trivial factor $\lambda$ the SSpj and GNpj algorithms have the same stability polynomial if we identify the parameters in the reference [2] 'β-p' notation as

$$\theta_0 = 1 = \beta_p, \ \theta_1 = \beta_{p-1}, \quad \theta_2 = \beta_{p-2}, \ldots, \ \theta_p = \beta_0 \quad . \tag{25}$$

The two algorithms then have the same stability conditions. This result also means that the two algorithms then correspond to the same p-step method. This can be seen by considering the p-step method

$$\sum_{j=0}^{p} (m\alpha_j + c\Delta t\gamma_j + k\Delta t^2\beta_j)u_{n+j} = 0 \tag{26}$$

applied to equation (3), with notation as used in references [3,4].

The stability polynomial of the p-step method given by equation (26) is

$$\sum_{j=0}^{p} (m\alpha_j + c\Delta t\gamma_j + k\Delta t^2\beta_j)\lambda^j = 0 \quad . \tag{27}$$

Comparing equation (27) with the equation giving the stability polynomial of SSp2 i.e. from equation (13):

$$mS_1 + c\Delta tS_2 + k\Delta t^2 S_3 = 0 \tag{28}$$

we see that we can identify

$$S_1 = \sum_{j=0}^{p} \alpha_j\lambda^j, \quad S_2 = \sum_{j=0}^{p} \gamma_j\lambda^j, \quad S_3 = \sum_{j=0}^{p} \beta_j\lambda^j \quad . \tag{29}$$

Hence obtaining the coefficients of the powers of $\lambda$ in the expansions of the determinants $S_1$, $S_2$, $S_3$ gives immediately the coefficients in the p-step method corresponding to SSp2. Since we have shown that with the conditions (25) on the parameters, GNp2 has precisely the same stability polynomial (apart from the trivial factor $\lambda$) as SSp2 these two single-step algorithms must then correspond to precisely the same p-step algorithm. The corresponding p-step algorithms can also be obtained by the method given by Osborne [5] or by the method in reference [2] but the method described here seems to be the simplest in the present context. By putting $m = 0$ we also have the corresponding results for SSp1 and GNp1 applied to

$$c\dot{u} + ku = 0 \quad . \tag{30}$$

REFERENCES

[1] O.C. ZIENKIEWICZ, W.L. WOOD, N.W. HINE & R.L. TAYLOR 'A unified set of single step algorithms, Part 1: general formulation and applications', Int. J. Num. Meth. Eng. 20, 1529-1552 (1984).

[2] M.G. KATONA & O.C. ZIENKIEWICZ 'A unified set of single step algorithms Part 3 : The beta-m method, a generalization of the Newmark scheme', Int. J. Num. Meth. Eng. 21, 1345-1359 (1985).

[3] W.L. WOOD 'Numerical integration of structural dynamics equations including natural damping and periodic forcing terms', Int. J. Num. Meth. Eng. 17, 281-289 (1981).

[4] A unified set of single step algorithms Part 4: Backward error analysis applied to the solution of the dynamic vibration equation. To appear in Int. J. Num. Meth. Eng.

[5] M.R. OSBORNE 'On Nordsieck's method for the numerical solution of ordinary differential equations', BIT 6, 51-57 (1966).