Çankaya University Journal of Science and Engineering
Volume 8 (2011), No. 1, 111–121

On Solving Coullet System by Differential Transformation Method

Mehmet Merdan[1,*], Ahmet Gökdoğan[2] and Vedat Suat Ertürk[3]

[1] *Gümüşhane University, Department of Geomatics Engineering, 29100 Gümüşhane, Turkey*
[2] *Gümüşhane University, Department of Software Engineering, 29100 Gümüşhane, Turkey*
[3] *Ondokuz Mayıs University, Department of Mathematics, 55139 Samsun, Turkey*
[*] *Corresponding author: mmerdan@gumushane.edu.tr*

Özet. Coullet sistemi olarak bilinen nonlineer denklem sisteminin çözümü diferensiyel dönüşüm yöntemi ile elde edildi. Sayısal sonuçlar, önerilen yöntemin etkinliğini ve doğruluğunu göstermek için dördüncü mertebeden Runge-Kutta yöntemi ile karşılaştırıldı. Önerilen yöntemin güçlü, doğru ve kolayca uygulanabilirliği gösterildi.

Anahtar Kelimeler. Coullet sistemi, diferensiyel dönüşüm, Runge-Kutta, nümerik metot.

Abstract. The differential transformation method is employed to solve a system of nonlinear differential equations, namely Coullet system. Numerical results are compared to those obtained by the fourth-order Runge-Kutta method to illustrate the preciseness and effectiveness of the proposed method. It is shown that the proposed method is robust, accurate and easy to apply.

Keywords. Coullet system, differential transformation, Runge-Kutta, numerical method.

1. Introduction

The concept of the differential transformation method has been introduced to solve linear and nonlinear initial value problems in electric circuit analysis [1-6]. The differential transformation method is a semi-numerical-analytic technique that formalizes the Taylor series in a totally different manner. With this method, the given differential equation and related initial conditions are transformed into a recurrence equation that finally leads to the solution of a system of algebraic equations as coefficients of a power series solution. This method is useful for obtaining exact and approximate solutions of linear and nonlinear differential equations. There is no need for linearization or perturbations, large computational work and round-off errors are avoided.

Received December 27, 2010; accepted May 15, 2011.

ISSN 1309 - 6788 © 2011 Çankaya University

The differential transformation method has solution in the form of polynomials. It is different from the traditional high order Taylor series method, which requires symbolic computation of the necessary derivatives of the data functions. The Taylor series method is computationally taken a long time for large orders. The present method reduces the size of computational domain and applicable to many problems easily. When dealing with non-linear systems of ordinary differential equations, it is often the case that a closed form analytic solution for the system of interest is normally unobtainable. In the absence of such solution, the accuracy of the differential transformation method is then usually tested with the classical numerical methods such as the fourth-order Runge-Kutta method [7]. Here the system we are interest in is Coullet system [8-13]. As is well-known, Coullet system does not admit a closed form solution and moreover it can exhibit chaotic behaviour for distinct parameter values:

$$\frac{dx}{dt} = y, \tag{1}$$

$$\frac{dy}{dt} = z, \tag{2}$$

$$\frac{dz}{dt} = cz + by + ax + dx^3, \tag{3}$$

where x, y, z are the state variables, and a, b, c and d are real constants. If the parameters are taken as $a = 0.8$, $b = -1.1$, $c = -0.45$ and $d = -1.0$, the system (1)-(3) exhibits chaotic dynamics.

2. Numerical Results

Taking the differential transformation of Equations (1)-(3) with respect to time t gives

$$X(k+1) = \frac{H}{k+1} Y(k), \tag{4}$$

$$Y(k+1) = \frac{H}{k+1} Z(k), \tag{5}$$

$$\begin{aligned} Z(k+1) = \frac{H}{k+1} \Bigg[& aX(k) + bY(k) + cZ(k) \\ & + d \sum_{k_2=0}^{k} \sum_{k_1=0}^{k_2} X(k_1) X(k_2 - k_1) X(k - k_2), \Bigg] \end{aligned} \tag{6}$$

where $X(k)$, $Y(k)$ and $Z(k)$ are the differential transformations of the corresponding functions $x(t)$, $y(t)$ and $z(t)$, respectively, and the initial conditions are given by $X(0) = 0.1$, $Y(0) = 0.41$ and $Z(0) = 0.31$.

The difference equations presented in Equations (4)-(6) describe Coullet system, from a process of inverse differential transformation, i.e.

$$x_i(t) = \sum_{k=0}^{n} \left(\frac{t}{H_i}\right)^k X_i(k), \ 0 \le t \le H_i, \tag{7}$$

$$y_i(t) = \sum_{k=0}^{n} \left(\frac{t}{H_i}\right)^k Y_i(k), \ 0 \le t \le H_i, \tag{8}$$

$$z_i(t) = \sum_{k=0}^{n} \left(\frac{t}{H_i}\right)^k Z_i(k), \ 0 \le t \le H_i, \tag{9}$$

where $k = 0, 1, 2, \ldots, n$ represents the number of terms of the power series, $i = 0, 1, 2, \ldots$, expresses the ith sub-domain and H_i is the sub-domain interval.

The accuracy of the DTM is demonstrated against Maples built-in fourth-order Runge Kutta procedure RK for the solutions of Coullet system. The domain is divided using $\Delta t = 0.01$ comparing with RK4 with step size $h = 0.001$. Figure 1 presents the comparison between DTM solution and RK4 solution. We can see the good agreement for DTM solution with RK4 solution. The phase portray of the Coullet system is given in Figure 2. It is clear that this is chaotic attractor for Coullet system. Also, Figure 3 shows the chaotic attractors for the Coullet system (1)-(3) using the DTM solution.The difference between 3-term DTM with $\Delta t = 0.01$ and RK4 with $h = 0.001$ is given in Figure 4. Figure 4 shows that the DTM has higher accuracy of the solution since in the x, y and z axis we have error until 10^{-5} (i.e., the solution via the new method has agreement with the purely numerical until 5 digit). In Equations (7)-(9), it must be indicated that the value of n selected equal to 3.

For the benefit of the reader, Maple codes regarding Figure 1(a), Figure 2, Figure 3(a) and Figure 4(a) are given in the Appendix, respectively.

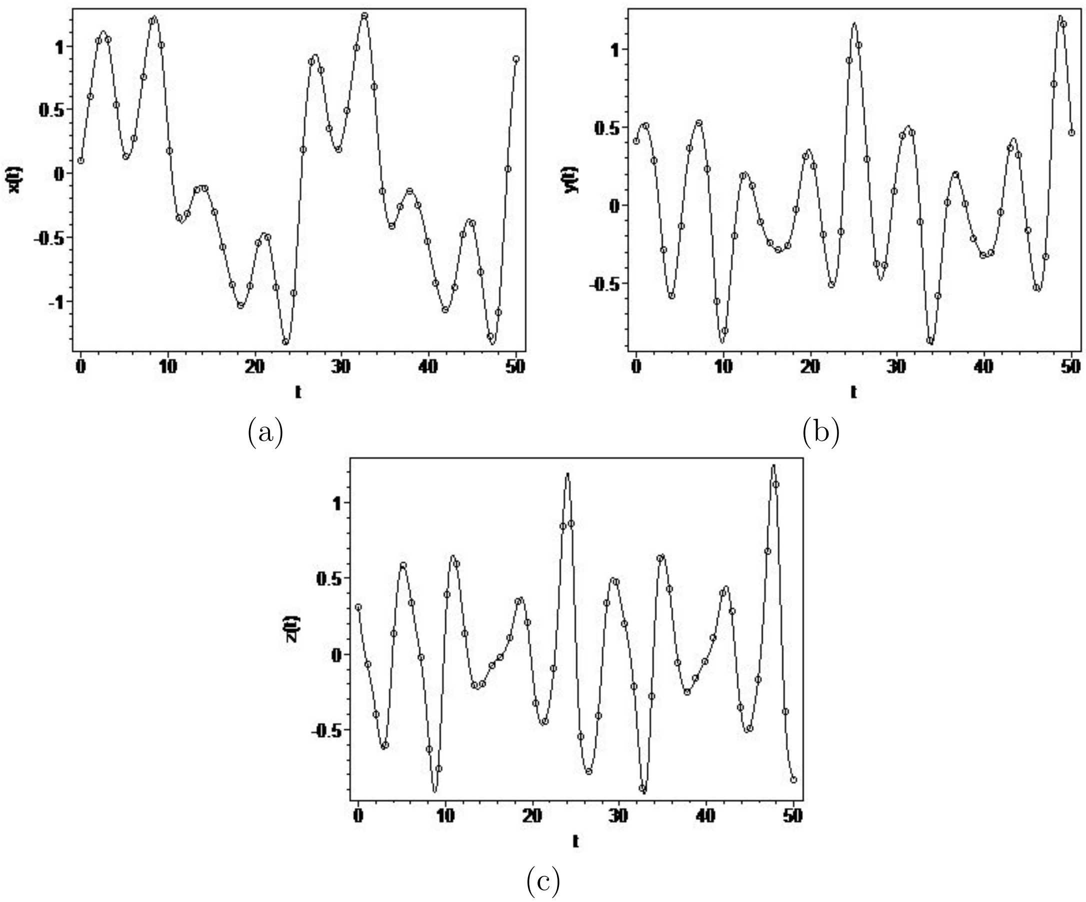


FIGURE 1. The DTM solution comparing with RK4: DTM (line); RK4 (circle).

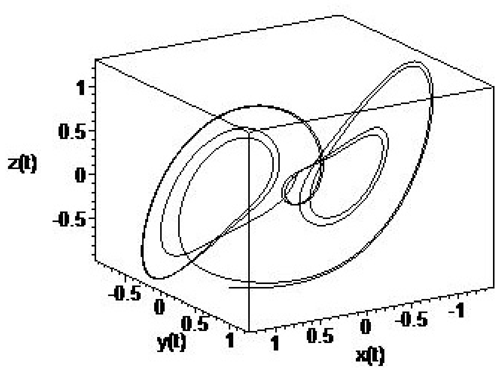
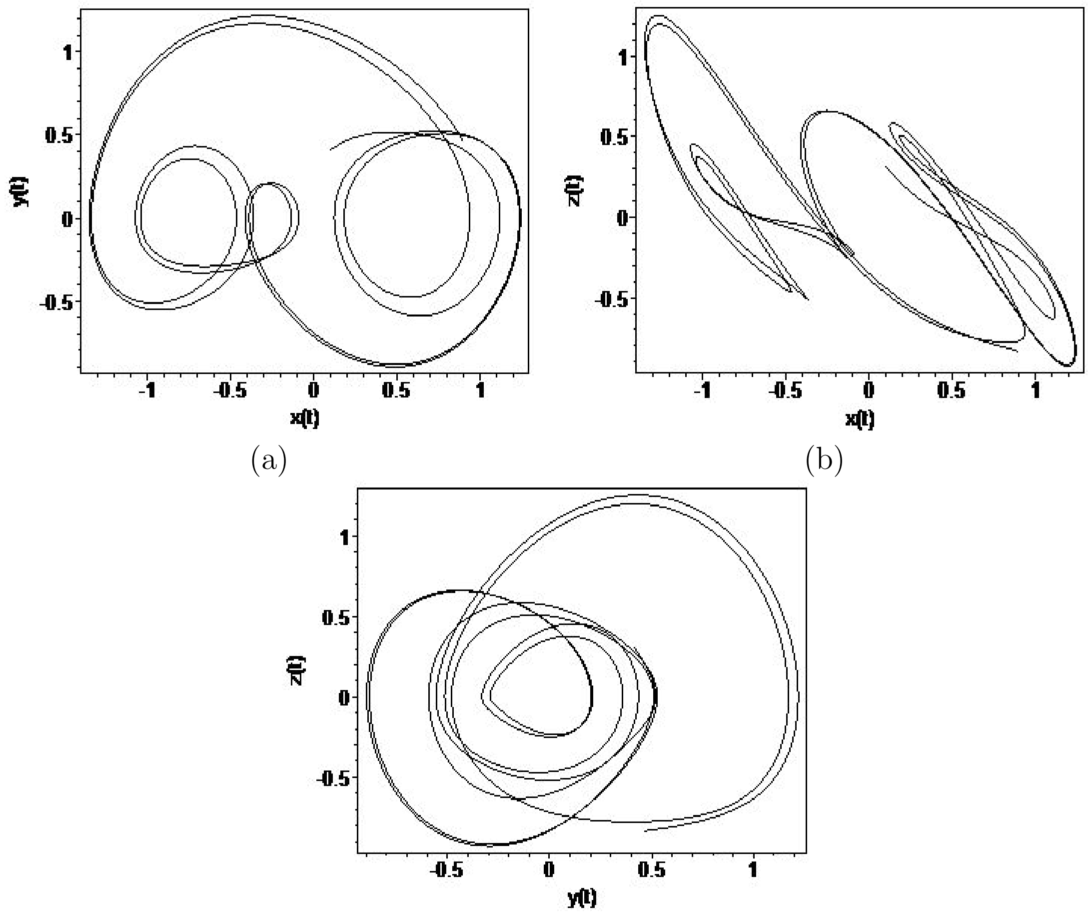


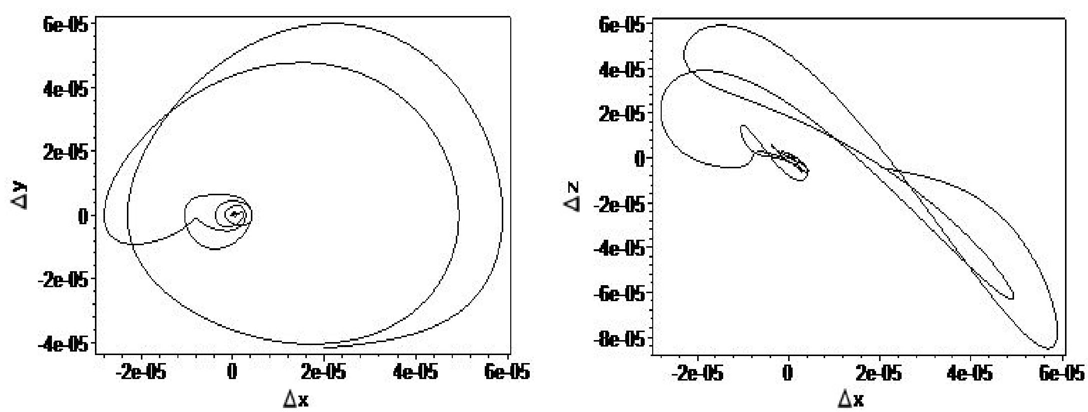
FIGURE 2. Phase portray for Coullet system with time span [0,50] using DTM.



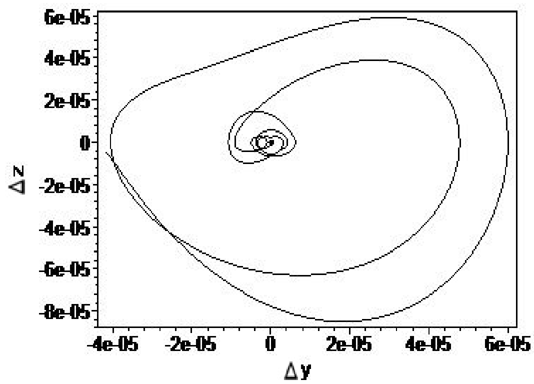
(a) (b)

(c)

FIGURE 3. Chaotic attractors for the system (1)-(3).



(a) (b)



(c)

FIGURE 4. Difference between DTM with $\Delta t = 0.01$ and RK4 with $h = 0.001$.

3. Conclusion

In this paper, DTM is implemented to solve Coullet chaotic system. Higher accuracy solution was obtained via this method. Comparison between DTM solution and RK4 solution is discussed and plotted. The solution via DTM is continuous on this domain and analytical at each subdomain which is the best in our knowledge.

Appendix

The code for Figure 1(a):

```
> restart;
> st:= time():
> N:=3: K:=5: eps:=0.0001: tson:=100:
> b1:=0.1: b2:=0.41: b3:=0.31:
> X(0):=b1: Y(0):=b2: Z(0):=b3: T(0):=0:
> i:=1:dt:=0.001:
> a:=0.8:b:=-1.1:c:=-0.45:d:=-1:
> for k from 0 to N do
> X(k+1):=(Y(k))/(k+1);
> Y(k+1):=(Z(k))/(k+1);
> Z(k+1):=(c*Z(k)+b*Y(k)+a*X(k)+d*sum(sum(X(k1)*X(k2-k1)*X(k-k2),
k1=0..k2),k2=0..k))/(k+1);
> end do:
```

```
> x:=sum(X(kk)*t^kk,kk=0..N):
> y:=sum(Y(kk)*t^kk,kk=0..N):
> z:=sum(Z(kk)*t^kk,kk=0..N):
> XX(0):=x: YY(0):=y:
> X(0):=subs(t=T(i),x);Y(0):=subs(t=T(i),y);Z(0):=subs(t=T(i),z);
> for k from 0 to N do
> X(k+1):=Y(k)/(k+1);
> Y(k+1):=Z(k)/(k+1);
> Z(k+1):=(c*Z(k)+b*Y(k)+a*X(k)+d*sum(sum(X(k1)*X(k2-k1)*X(k-k2),
k1=0..k2),k2=0..k))/(k+1);
> end do:
> x:=sum(X(kk)*(t-T(i))^kk,kk=0..N):
> y:=sum(Y(kk)*(t-T(i))^kk,kk=0..N):
> z:=sum(Z(kk)*(t-T(i))^kk,kk=0..N):
> XX(i):=x;YY(i):=y;ZZ(i):=z;
> i:=i+1;
> end do:
> time() - st;
> i;
> sys:=diff(xx(t),t)=yy(t), diff(yy(t),t)=zz(t),diff(zz(t),t)
=c*zz(t)+b*yy(t)+a*xx(t)+d*xx(t)^3: fcns := {xx(t),yy(t),zz(t)}:
> dsol1:=dsolve({sys,xx(0)=b1,yy(0)=b2,zz(0)=b3},fcns,type=numeric,
output=listprocedure,stepsize=0.001,method=classical[rk4]):
> with(plots):
> T(i):=tson:
> H1:=plot(XX(1),t=T(1)..T(2),color=black):
> P1:={H1}:
> for k from 2 to i-1 do
> H1:=plot(XX(k),t=T(k)..T(k+1),color=black):
> P1:={op(P1),H1};
> end do:
> H2:=odeplot(dsol1,[t,xx(t)],0..tson,style=point,symbol=circle,
color=black):
> P1:={op(P1),H2}:
> display(P1,labels=["t","x(t)"],title="(a3)");
```

```
> with(plots):T(i):=tson:
> H1:=plot(YY(1),t=T(1)..T(2),color=black):
> P1:={H1}:
> for k from 2 to i-1 do
> H1:=plot(YY(k),t=T(k)..T(k+1),color=black):
> P1:={op(P1),H1};
> end do:
> H2:=odeplot(dsol1,[t,yy(t)],0..tson,style=point,symbol=circle,
color=black):
> P1:={op(P1),H2}:
> display(P1,labels=["t","y(t)"],title="(b3)");
> with(plots):T(i):=tson:
> H1:=plot(ZZ(1),t=T(1)..T(2),color=black):
> P1:={H1}:
> for k from 2 to i-1 do
> H1:=plot(ZZ(k),t=T(k)..T(k+1),color=black):
> P1:={op(P1),H1};
> end do:
> H2:=odeplot(dsol1,[t,zz(t)],0..tson,style=point,symbol=circle,
color=black):
> P1:={op(P1),H2}:
> display(P1,labels=["t","z(t)"],title="(c3)");
```

The code for Figure 2:

```
  > with(plots):
> T(i):=tson:
> H1:=spacecurve({[XX(1),YY(1),ZZ(1)]},t=T(k)..T(k+1),color=black):
> P1:={H1}:
> for k from 2 to i-1 do
> H1:=spacecurve({[XX(k),YY(k),ZZ(k)]},t=T(k)..T(k+1),color=black):
> P1:={op(P1),H1};
> end do:
> display(P1,labels=["x","y","z"]);
```

The code for Figure 3(a):

```
with(plots):T(i):=tson:
> H1:=plot([XX(1),YY(1),t=T(1)..T(2)],color=black):
> P1:={H1}:
> for k from 2 to i-1 do
> H1:=plot([XX(k),YY(k),t=T(k)..T(k+1)],color=black):
> P1:={op(P1),H1};
> end do:
> display(P1,labels=["x","y"]);

> with(plots):
> T(i):=tson:
> H1:=plot([XX(1),ZZ(1),t=T(1)..T(2)],color=black):
> P1:={H1}:
> for k from 2 to i-1 do
> H1:=plot([XX(k),ZZ(k),t=T(k)..T(k+1)],color=black):
> P1:={op(P1),H1};
> end do:
> display(P1,labels=["x","z"]);

> with(plots):
> T(i):=tson:
> H1:=plot([YY(1),ZZ(1),t=T(1)..T(2)],color=black):
> P1:={H1}:
> for k from 2 to i-1 do
> H1:=plot([YY(k),ZZ(k),t=T(k)..T(k+1)],color=black):
> P1:={op(P1),H1};
> end do:
> display(P1,labels=["y","z"]);
```

The code for Figure 4(a):

```
> byK:=10:
> for i from 0 to K*NN by byK do
> dsol1x := subs(dsol1,xx(t)):
> dsol1y := subs(dsol1,yy(t)):
```

```
> dsol1z := subs(dsol1,zz(t)):
> deltaxx(i/byK):=(dsol1x(i*dt/K)-XX(i));
> deltayy(i/byK):=(dsol1y(i*dt/K)-YY(i));
> deltazz(i/byK):=(dsol1z(i*dt/K)-ZZ(i));
> ddxx(i/byK):=(XX(i));
> ddyy(i/byK):=(YY(i));
> ddzz(i/byK):=(ZZ(i));
> end do:
> plot([seq( [deltaxx(i),deltayy(i)], i=0..K*NN/byK )],color=black,
labels=["x","y"],labeldirections=[horizontal,vertical]);
> plot([seq( [deltaxx(i),deltazz(i)], i=0..K*NN/byK )],color=black,
labels=["x","z"],labeldirections=[horizontal,vertical]);
> plot([seq( [deltayy(i),deltazz(i)], i=0..K*NN/byK )],color=black,
labels=["y","z"],labeldirections=[horizontal,vertical]);
```

References

[1] J. K. Zhou, *Differential Transformation and its Applications for Electrical Circuits (In Chinese)*, Huazhong University Press, Wuhan, China 1986.

[2] V. S. Ertürk, Differential transformation method for solving differential equations of Lane-Emden type, *Mathematical and Computational Applications* **12** (2007), 135–139.

[3] V. S. Ertürk, Solution of linear twelfth-order boundary value problems by using differential transform method, *International Journal of Applied Mathematics & Statistics* **13**(M08) (2008), 57–63.

[4] S.-H. Chang and I.-L. Chang, A new algorithm for calculating one-dimensional differential transform of nonlinear functions, *Applied Mathematics and Computation* **195** (2008), 799–808.

[5] H. Demir and İ. Ç. Süngü, Numerical solution of a class of nonlinear Emden-Fowler equations by using differential transform method, *Çankaya Üniversitesi Journal of Arts and Sciences* **12** (2009), 75–81.

[6] M. Merdan and A. Gökdoğan, Solution of nonlinear oscillators with fractional nonlinearities by using the modified differential transformation method, *Mathematical and Computational Applications* **16** (2011), 761–772.

[7] I. Hashim, M. S. M. Noorani, R. Ahmad, S. A. Bakar, E. S. Ismail and A. M. Zakaria, Accuracy of the Adomian decomposition method applied to the Lorenz system, *Chaos, Solitons & Fractals* **28** (2006), 1149–1158.

[8] A. Arneodo, P. Coullet and C. Tresser, Possible new strange attractors with spiral structure, *Communications in Mathematical Physics* **79** (1981), 573–579.

[9] P. Coullet, C. Tresser and A. Arneodo, Transition to stochasticity for a class of forced oscillators, *Physics Letters A* **72** (1979), 268–270.

[10] J.-B. Hu, Y. Han and L.-D. Zhao, Synchronization in the Genesio Tesi and Coullet systems using the backstepping approach, *Journal of Physics: Conference Series* **96** (2008), 012150.

[11] X. Shi and Z. Wang, Adaptive synchronization of Coullet systems with mismatched parameters based on feedback controllers, *International Journal of Nonlinear Science* **8** (2009), 201–205.

[12] J. Ghasemi, A. Ranjbar N. and A. Afzalian, Synchronization in the Genesio-Tesi and Coullet systems using the sliding mode control, *International Journal of Engineering* **4** (2010), 60–65.

[13] L. Yang-Zheng and F. Shu-Min, Synchronization in the Genesio-Tesi and Coullet systems with nonlinear feedback controlling, *Acta Physica Sinica (Chinese Edition)* **54** (2005), 3490–3495.