

# ELSŐ- ÉS MÁSODRENDŰ IDŐBELI DISZKRETIZÁCIÓ KÖRHENGER KÖRÜLI ÁRAMLÁS ESETÉN

## FIRST- AND SECOND-ORDER TEMPORAL DISCRETIZATION FOR COMPUTATION OF FLOW AROUND A CIRCULAR CYLINDER

dr. Baranyi László<sup>1</sup>, Daróczy László<sup>2</sup>

### ABSTRACT

*This paper deals with the two-dimensional numerical simulation of low-Reynolds number flow past a stationary circular cylinder using the finite difference method. We investigate the effect of temporal discretization (1<sup>st</sup> order Euler and 2<sup>nd</sup> order Runge-Kutta) on force coefficients and Strouhal number. Additionally, solvers for two types of hardware: CPU and GPGPU (General-Purpose computing on Graphics Processing Units) are used for validation of the code. Computations were carried out for Reynolds numbers 100 and 150, for different dimensionless time steps (0.0001; 0.0002; 0.0004; 0.0005) and at different mesh sizes (512x450; 360x260).*

*Computational results obtained for the 1<sup>st</sup> and 2<sup>nd</sup> order methods agree well using both CPU and GPGPU, though the latter is much faster. Results also compare well with values in the literature. The 2<sup>nd</sup> order method is generally considered better, but its advantage of high accuracy at larger time steps cannot be utilized here, since the code demands relatively small time steps (it diverges at larger time steps due to using successive over-relaxation). Results obtained with 1<sup>st</sup> order Euler discretization proved to be equally accurate in this case.*

### 1. BEVEZETÉS

A különféle testek körüli áramlások nagy pontosságú szimulációja során nagyon fontos egy megfelelő egyensúlyt találnunk a számítási idő és a pontosság között. A magasabbrendű numerikus eljárások alkalmazása sem kivétel ez alól, mivel bár azonos idő- és térbeli diszkretizáció alkalmazása esetén a magasabb

rendű eljárások pontosabbak, azonban magasabb a számítás igényük is. Másképpen megfogalmazva ugyanolyan pontosság érhető el kisebb méretű háló vagy nagyobb időlépcső esetén is, amennyiben a magasabbrendű eljárásokat alkalmazzuk.

A kereskedelmi szoftverekben, de talán még sok kutatásban is legelterjedtebben alkalmazott eljárás a sok tekintetben már elavultnak tekintett elsőrendű Euler eljárás. A másodrendű eljárások között a legelterjedtebb a sokféle változatban létező Runge-Kutta (RK) illetve Adams-Bashforth eljárások [1]. A még magasabb rendű módszerek közül leggyakrabban a negyedrendű Runge-Kutta eljárást használják.

A jelen tanulmány során a másod- és negyedrendű Runge-Kutta eljárást is megvizsgáltuk, bár ez utóbbi a probléma jelen megfogalmazása mellett sem a SOR (successive over-relaxation), sem a BiCGSTAB (Bilinear Conjugate Gradient STABILized method) eljárás mellett nem bizonyult hatékonynak a rendkívül nagy futási idő miatt. A másodrendű Runge-Kutta (RK2) módszereken belül a Heun, Midpoint, Ralston és Taylor sorozatok [2, 3] módszerei közül végül a Heun eljárást választottuk ki, majd hasonlítottuk össze az Euler-eljárás eredményeivel különböző időlépcsők, hálóméretetek, Reynolds számok és számítási hardver esetén a két-dimenziós (2D), kis Reynolds számú, homogén párhuzamos áramlásba helyezett körhenger körüli áramlás esetén [4].

### 2. SZÁMÍTÁSI ELJÁRÁS

A számításokat egy saját fejlesztésű, véges differenciák módszerén alapuló szoftver segítségével végeztük el, melyet a kis Reynolds számú síkbeli, álló vagy rezgő mozgást végző henger körüli áramlások vizsgálatára fejlesztettünk ki.

Az eljárás az összenyomhatatlan, konstans tulajdonságokkal rendelkező newtoni folyadékokra felírt dimenziótlant Navier-Stokes egyenletet, a kontinuitási egyenletet, valamint a nyomásra felírt Poisson egyenletet oldja meg, [5].

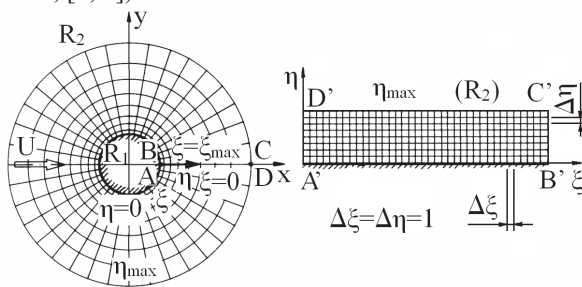
<sup>1</sup>egyetemi tanár, Miskolci Egyetem, Áramlás és Hőtechnikai Gépek Tanszéke

<sup>2</sup>gépészmérnök MSc, Miskolci Egyetem

A tényleges fizikai tartomány két,  $R_1$  (körhenger fala) és  $R_2$  (számítási tartomány külső pereme) dimenziótlan sugarú koncentrikus kör közötti terület (l. 1. ábra).  $R_1$  mentén a sebesség zéró, a nyomásra pedig egy Neumann típusú peremfeltételt használunk, míg  $R_2$  mentén potenciáláramlást tételezünk fel, [5].

A peremfeltételek pontos kielégítése érdekében peremre illeszkedő görbevonaltú koordináta-rendszert használunk. A fizikai tartomány logaritmikusan diszkrétizált hálóját egy téglalap alakú, ekvidisztáns felosztású számítási tartományra képezzük le (l. 1. ábra), majd a transzformált egyenleteket véges differenciák módszerével oldjuk meg, [6, 7].

A térkoordináták szerinti deriváltakat negyedrendű centrális differenciákkal közelítjük, kivéve a konvektív tagot, amelyet egy harmadrendű módosított „upwind” sémával kezelünk. A mozgásegyenletet időben explicit módon integráljuk, nyomásra vonatkozó Poisson egyenletet a SOR eljárással oldjuk meg, miközben a kontinuitási egyenletet is kielégítjük. További részletek az [5, 6] dolgozatokban találhatóak. A második szerző által kifejlesztett 2D FORTRAN kód alapos vizsgálatnak lett alávetve az évek során, és eredményei nagyon jól megegyeznek a szakirodalmi kísérleti és számítási eredményekkel mind álló, mind rezgő henger esetén, [6, 7].



1. ábra. Fizikai és számítási tartomány

### 3. EULER ÉS RUNGE-KUTTA IDŐLÉPTETÉSEK

Nézzük meg először az egyenletek megoldásának menetét a Runge-Kutta eljárás alkalmazásával! A rezgő hengerhez rögzített, nem-inerciarendszerben felírt dimenziótlan Navier-Stokes egyenlet:

$$\frac{\partial \mathbf{v}}{\partial t} = -(\mathbf{v} \cdot \nabla) \mathbf{v} - \nabla p + \frac{1}{Re} \nabla^2 \mathbf{v} - \mathbf{a}_0, \quad (1)$$

ahol  $\mathbf{v}=\{u,v\}$  a sebességvektor,  $p$  a nyomás,  $Re=Ud/\nu$  a Reynolds szám,  $U$  a párhuzamos áramlás sebessége,  $d$  a henger átmérője,  $\nu$  a folyadék kinematikai viszkozitása,  $\mathbf{a}_0$  a henger gyorsulása,  $t$  az idő; minden változó dimenziótlan formában felírva [5,6]. A nyomásra vonatkozó Poisson egyenlet [6]:

$$\frac{\partial D}{\partial t} = 2 \left[ \frac{\partial u}{\partial x} \frac{\partial v}{\partial y} - \frac{\partial u}{\partial y} \frac{\partial v}{\partial x} \right] - \nabla^2 p. \quad (2)$$

Az  $n$ . időlépcsőben a  $\mathbf{v}^n, p^n$  megoldások ismeretében az (1) egyenlet idő szerinti integrálása Heun RK2 módszere alapján az alábbiak szerint történik.

**1. lépés.** A  $\mathbf{k}_1$  Runge-Kutta tényező meghatározása:

$$\mathbf{k}_1 = - \left[ (\mathbf{v}^n \cdot \nabla) \mathbf{v}^n + \nabla p^n - \frac{1}{Re} \nabla^2 \mathbf{v}^n \right]. \quad (3)$$

Ezt felhasználva meghatározhatjuk a  $\mathbf{v}^*=\{u^*,v^*\}$  átmeneti sebességet

$$\mathbf{v}^* = \mathbf{v}^n + \mathbf{k}_1 \Delta t - \mathbf{v}_0^n + \mathbf{v}_0^{n+1}, \quad (4)$$

ahol  $\mathbf{v}_0$  a henger sebessége és  $\Delta t$  az időlépcső. Ezek alapján, (2) felhasználásával meghatározható a  $p^*$  átmeneti nyomás

$$\frac{0-D^n}{\Delta t} = 2 \left[ \frac{\partial u^*}{\partial x} \frac{\partial v^*}{\partial y} - \frac{\partial u^*}{\partial y} \frac{\partial v^*}{\partial x} \right] - \nabla^2 p^*, \quad (5)$$

ahol  $D^n = \text{div}(\mathbf{u}^n)$ . Így az (5) Poisson egyenlet megoldásával párhuzamosan a  $D^*=\text{div}(\mathbf{v}^*)=0$  kontinuitási egyenletet is kielégítjük.

**2. lépés.** A  $\mathbf{k}_2$  Runge-Kutta tényező meghatározása:

$$\mathbf{k}_2 = - \left[ (\mathbf{v}^* \cdot \nabla) \mathbf{v}^* + \nabla p^* - \frac{1}{Re} \nabla^2 \mathbf{v}^* \right]. \quad (6)$$

A  $\mathbf{k}_1, \mathbf{k}_2$  tényezők ismeretében kapjuk a következő  $(n+1)$  időlépcsőhöz tartozó  $\mathbf{v}^{n+1}=\{u^{n+1},v^{n+1}\}$  sebességet, illetve a  $p^{n+1}$  nyomást:

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \frac{1}{2}(\mathbf{k}_1 + \mathbf{k}_2) \Delta t - \mathbf{v}_0^n + \mathbf{v}_0^{n+1}, \quad (7)$$

$$\frac{0-D^n}{\Delta t} = 2 \left[ \frac{\partial u^{n+1}}{\partial x} \frac{\partial v^{n+1}}{\partial y} - \frac{\partial u^{n+1}}{\partial y} \frac{\partial v^{n+1}}{\partial x} \right] - \nabla^2 p^{n+1}. \quad (8)$$

Amennyiben a fenti két lépcsőből csak az elsőo hajtjuk végre, akkor az Euler módszert kapjuk. A fentiekből látható, hogy az RK2 módszernél az időigényes SOR eljárást kétszer is végre kell hajtani egyetlen időlépcsőn belül, míg az Eulernél csak egyszer. A számítások folyamán azt tapasztaltuk, hogy a futásidő ennek ellenére nem duplázódik meg, mivel a SOR eljárás a 2. lépés során gyorsabban konvergál.

### 4. A HASZNÁLT SZOFTVER

A jelen tanulmányban az első szerző által kifejlesztett FlowCFD nevű program segítségével végeztük el a számításokat. A FlowCFD egy C++ nyelven megírt, körhenger körüli áramlások vizsgálatára kidolgozott, [5-7] algoritmusán alapuló szoftvercsomag, amely lehetővé teszi a szimulációk párhuzamos ütemezését a processzoron és grafikai kártyákon, valamint kiterjedt poszt-processzálási lehetőséget biztosít.

A szoftver számítási motorja négy különböző számítási eljárást tartalmaz. Az első változat a CPU-n fut Euler-típusú időléptetést alkalmazva, ez a kód a második szerző által korábban kifejlesztett FORTRAN program (l. [6]) C++ nyelvre történő átirása. A FORTRAN és C++ program változatok teljes

mértékben megegyező eredményt adnak, amely megfelelő validálást nyújt ezen számítási eljárás számára, hiszen az eredeti FORTRAN programot a publikációk során alaposan teszteltük, [7]. A második változat a grafikai kártyák nyújtotta hatalmas számítási teljesítményből profitál a GPGPU technika révén, akár 8-20x gyorsulást is elérve, de továbbra is az Euler-típusú időléptetést alkalmazva. A kernelek itt NVIDIA® CUDA™ C nyelven vannak megírva, de az eredeti SOR eljárás helyett a vörös-fekete Gauss-Seidel SOR módszert alkalmazza [8], amely jól párhuzamosítható.

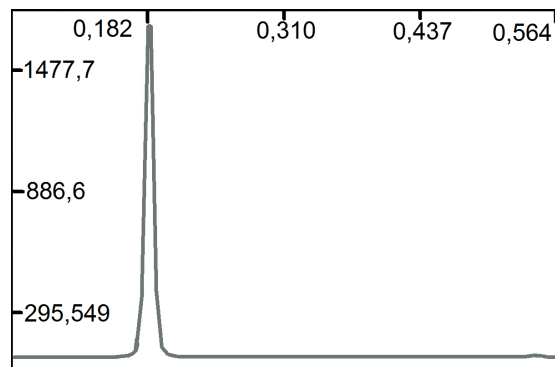
A későbbiekben a programot kiegészítettük az Euler mellett másodrendű Runge-Kutta időléptetéssel is, mind a processzoron (harmadik változat), mint a grafikai kártyán (negyedik változat).

A processzoron végzett számításokat egy Intel® Core i7™ 980X processzoron, míg a GPU-n végzett számításokat egy Tesla™ C2050 GPU 448 magján, duplapontossággal és ECC (error correcting code memory) beállítással futtattuk le.

## 5. VIZSGÁLT ESETEK, KIÉRTÉKELÉS

A szimulációkat álló hengerre vonatkozóan,  $Re=100$  és  $150$  esetén,  $\Delta t=(1,2,4,5)\times 10^{-4}$  dimenzióatlanított időlépcsővel, minimum  $t=400$  dimenzióatlan időtartamig számítottuk. A  $\Delta t\geq 6\times 10^{-4}$  esetén a SOR eljárás nem konvergált, így a SOR és RK eljárások valamilyen fokon inkompatibilisek a számítási eljárásban.

A CPU és GPU számítások során a számítási tartományt  $512\times 450$  (kerület x radiális) háló segítségével diszkretizáltuk ( $R_2/R_1=160$ ). Emellett a CPU-n  $Re=100$  esetén még egy kisebb,  $360\times 260$  ( $R_2/R_1=60$ ) elemet tartalmazó hálót is megvizsgáltunk.



2. ábra. FFT frekvenciaspektrum ( $Re=150$ ,  $C_L$ )

Az adatok feldolgozásánál igen fontos a periódusidő pontos meghatározása, mivel ez alapján számítható a felhajtó- és ellenállás-tényező időátlag illetve rms (root-mean-square) értéke. A gyors Fourier transzformáció (FFT) minden esetben egyetlen frekvenciacsúcsot talált a frekvencia-spektrumon (ld. 2. ábra), így az adatok kiértékeléséhez szükséges periódusidő egyértelműen meghatározható. A program ezt az értéket a beépített

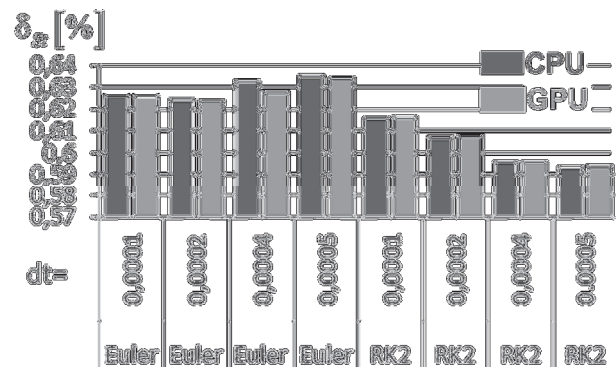
MaxCut algoritmussal állapította meg, mely az egyes tényezők lokális maximumainak elhelyezkedése alapján határozza meg a periódusidő értékét nagy pontossággal.

## 6. EREDMÉNYEK

A vizsgálatok során meghatároztuk az ellenállás-tényező ( $C_D$ , átlag), felhajtóerő-tényező ( $C_L$  amplitúdó), hátsó nyomástényező ( $-C_{pb}$ , átlag) és a dimenzióatlan örvényleválási frekvencia, azaz a Strouhal szám ( $St=f/d/U$ ) értékeit. Ezeket hasonlítottuk össze a egyrészt a jelen tanulmány különböző beállításai alapján nyert, valamint Posdziech és Grundmann [9] eredményeivel, akik a spektrális elem módszer alkalmazásával egy olyan vizsgálatot végeztek az álló körhenger körüli 2D áramlásokra, amely során szisztematikusan változtatták a számítási tartomány méretét. Az összehasonlítás nagyfokú egyezést mutatott minden vizsgált esetben.

A fentiek illusztrálására szeretnénk itt bemutatni néhány példát. A  $C_L$  felhajtóerő-tényező amplitúdójára a legnagyobb számítási tartománya esetén [9]  $Re=100$  esetben  $0,3161$  értéket ad meg. A FlowCFD szoftver ugyanezen értékre a kisebb háló esetén  $0,3225$ - $0,3240$ , a nagyobb tartomány esetén pedig  $0,3197$ - $0,3204$  értékeket adott (időlépcső nagyságától függően). Az  $Re=150$  esetre [9]  $0,5030$ , míg a FlowCFD  $0,5093$ - $0,5105$  értéket adott, amely jó egyezésnek mondható.

A  $C_D$  ellenállás-tényezőre vonatkozóan [9]  $Re=100$  esetén  $1,3123$ ,  $Re=150$  esetén pedig  $1,2992$  értéket ad meg. Ugyanezen esetekre a FlowCFD  $1,3221$ - $1,3238$ , illetve  $1,3102$ - $1,3122$  értéktartományt adott mind a CPU, mind a GPU kódok esetén.



3. ábra. Strouhal szám – Relatív hiba ( $Re=100$ )

A Strouhal szám esetén  $Re=100$  esetén [9]  $0,1633$  (legnagyobb tartomány), míg  $Re=150$  esetén  $0,18366$  értéket ad meg (legkisebb hibát adó nem-egész polinom alapján). Az  $Re=100$  esetén a FlowCFD  $0,1642$ - $0,1643$  közötti értékeket adott meg a nagy ( $0,576$ - $0,636\%$  relatív hiba [9]-hoz képest, l. 3. ábra), és  $0,1652$ - $0,1654$ -ot a kisebb hálóra ( $1,169$ - $1,292\%$  relatív hiba). Az  $Re=150$  esetén a FlowCFD  $0,18356$ - $0,18365$  közötti értékeket adott ( $0,0006\%$ - $0,0518\%$  hiba).

Az elmondottak alapján látható, hogy az elsőrendű és másodrendű időbeli léptetések szinte azonos eredményt szolgáltatottak, azaz az alkalmazott paraméterek mellett az elsőrendű eljárás is képes volt megfelelni, a magasabb rendű eljárásokkal megegyező eredményt adni.

## 7. ÖSSZEFOGLALÁS

A vizsgált időlépcsők esetében az első és másodrendű eredmények is jó egyezést mutattak, azaz a vizsgált körülmények között az elsőrendű eredmények is elegendő pontosságot tudtak biztosítani. Ugyanakkor a kapott eredmények a spektrális elem módszer eredményeivel is megegyeztek [9].

Bár a Runge-Kutta eljárást sikerült implementálni a meglévő és jól-kipróbált kódba, azonban ennek ellenére képtelenek voltunk profitálni a magasabb rendű eljárás előnyeiből – azaz növelni az idő-lépcső értékét azonos pontosság esetén, ugyanis az időlépcső növelése esetén a SOR eljárás nem konvergált. Úgy tűnik, hogy jelen implementáción belül a SOR és RK eljárások inkompatibilisek valamilyen fokon, így a SOR eljárás lecserélésének lehetőségét már vizsgáljuk. Azonban az mindenképpen pozitív eredmény, hogy az alkalmazott paraméterek mellett az elsőrendű eljárás is ugyanazt a pontosságot biztosítja, mint a másodrendű eljárás.

Az eredmények alapján azt a következtetést is levonhatjuk, hogy a CPU és GPGPU kód eredményei között nem mutatkozik jelentős különbség (csak a definiált pontosságon belül,  $10^{-5}$ ), de a GPGPU kód 8-20x nagyobb sebességet nyújt.

## 8. SUMMARY

For the investigated time steps the 2nd order results agreed very well with the results of the 1st order temporal discretisation. Additionally, all the results agree well with the spectral element results given in [9].

Although we were able to implement the Runge-Kutta method into our existing and proven code, we found that we were unable to take advantage of the benefits of higher order methods, i.e., being able to use larger time steps and still obtain the same accuracy. When the time step was increased the SOR method failed to converge. It appears that in the current implementation the SOR method and the Runge-Kutta method are somewhat incompatible. We are currently investigating ways to replace SOR.

However it is a clearly positive result that the 1st order method is able to provide the same accuracy as the 2nd order method within the applied parameters. This confirms that the 1st order method yields accurate results under the conditions investigated.

While the CPU and GPGPU codes result in no significant differences (only up to the defined precision,  $10^{-5}$ ), the GPGPU version is 8-20 times faster.

## 9. KÖSZÖNETNYILVÁNÍTÁS

A szerzők köszönetüket fejezik ki a K 76085 sz. OTKA projekt keretében megvalósult támogatásért. A kutató munka a TÁMOP-4.2.1.B-10/2/KONV-2010-0001 jelű projekt részeként – az Új Magyarország Fejlesztési Terv keretében – az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg. Külön köszönet jár az NVIDIA™ részére, amiért az „Academic Partnership Program” keretében egy NVIDIA® Tesla™ C2050-t adományozott a kutatás támogatására.

## 10. IRODALOM

- [1] FERZIGER, J.H., PERIĆ, M.: *Computational Methods for Fluid Dynamics* (3rd edition). Springer-Verlag, Berlin, 2002.
- [2] CHEEVER, E.: The 2nd order Runge-Kutta method. <http://www.swarthmore.edu/NatSci/echeever1/Ref/NumericalInt/RK2.html>.
- [3] KAW, A.K., KALU, E.E.: Numerical Methods with Applications. Ed. World Wide Web, <http://www.autarkaw.com>, 2009.
- [4] DARÓCZY, L., BARANYI, L.: Euler and second-order Runge-Kutta methods for computation of flow around a cylinder, Proc. XXVI. microCAD International Scientific Conference, Section N, pp. 1-6, Paper No. N10, 2012.
- [5] BARANYI L.: Mozgó henger körüli lamináris áramlás vizsgálata. Miskolci Egyetem Habilitációs füzetek, 2007.
- [6] BARANYI, L.: Computation of unsteady momentum and heat transfer from a fixed circular cylinder in laminar flow. *Journal of Computational and Applied Mechanics* 4(1), 13-25, 2003.
- [7] BARANYI, L.: Numerical simulation of flow around an orbiting cylinder at different ellipticity values. *Journal of Fluids and Structures* 24, 883-906, 2008.
- [8] SHINN, A.F.: Computational Fluid Dynamics (CFD) using Graphics Processing Units. [http://impact2.crhc.illinois.edu/gpucourses/courses/slecture/CaseStudy\\_CFD\\_GPU.pdf](http://impact2.crhc.illinois.edu/gpucourses/courses/slecture/CaseStudy_CFD_GPU.pdf).
- [9] POSDZIECH, O., GRUNDMANN, R.: A systematic approach to the numerical calculation of fundamental quantities of the two-dimensional flow over a circular cylinder. *Journal of Fluids and Structures* 23, 479-499, 2007.