

Paper title

Tytuł artykułu w języku polskim

Tomasz Adam Kowalski^{a,*}, Dariusz Nowak^b^a Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland^b Another Institution, address

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem.

Keywords: stress; spring boot; dynamics simulation

Streszczenie

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem.

Słowa kluczowe: naprężenia; spring boot; dynamika molekularna

*Corresponding author

Email address: kowalski@company.com (T. A. Kowalski)

Published under Creative Common License (CC BY 4.0 Int.)

1. Wstęp

Niniejszy dokument opracowano dla autorów przygotowujących artykuły w języku polskim lub angielskim z zamiarem opublikowania w czasopiśmie Journal of Computer Sciences Institute (J. Comput. Sci. Inst.). Dokument należy wysłać drogą elektroniczną poprzez stronę wydawnictwa: <https://ph.pollub.pl/index.php/jcsi>.

Dokument w formacie dwukolumnowych składamy w oparciu o przykładowy szablon dostępny pod adresem: <https://cs.pollub.pl/oldjcsi/jcsi.zip>. W pliku umieszczono przykładowy dokument w formacie docx. Wygląd tworzonego dokumentu powinien być jak najbardziej zbliżony do przykładowego dokumentu. Dotyczy to bezwzględnie marginesów, interlinii, rozmiaru i stylu czcionek, pozycji i wyglądu podpisów nad/pod obiektami. Wzory, rysunki i inne obiekty numerowane powinny być numerowane automatycznie. Interlinia i czcionka powinna być taka sama w całym dokumencie. Bezwzględnie należy używać zdefiniowanych stylów do zmiany sposobu formatowania dokumentu. W dokumencie należy użyć czcionkę Times New Roman, 10 pkt z interlinią pojedynczą. Tekst powinien być wyjustowany z włączonym dzieleniem wyrazów. Marginesy powinny wynosić: górny i dolny 2,5cm, lewy i prawy 2cm, szerokość kolumn 8cm z odstępem 1cm. Pierwszy akapit po każdym rozdziale musi być bez wcięcia, w tym celu dodany został specjalny styl „Tekst podstawowy bez wcięcia”.

Po tytułach i danych autorów umieszczamy treść streszczeń w języku angielskim i polskim. Powinny być zgodne ze sobą i nie dłuższe niż 800 znaków (łącznie ze spacjami). Poniżej umieszczamy 2-4 słowa kluczowe

dla wersji angielskiej i polskiej, tak jak w przykładzie, które oddzielone są średnikiem. Potem umieszczamy treść dokumentu. Akapity występujące bezpośrednio na początku każdego rozdziału nie posiadają wcięcia. Zabrania się używania innych stylów. Finalną wersję pracy wysyłamy poprzez stronę www. Proszę wysłać do wydawnictwa plik źródłowy w formacie **nazwisko_autora_korespondencyjnego.docx** oraz zgodny plik PDF (*Portable Document Format*).

Redakcja zastrzega sobie możliwość zmiany ostatecznego wyglądu graficznego dokumentu.

2. Wzory

Nazwy stałych oraz liczby należy pisać czcionką prostą na przykład 3,1415hPa. Nazwy zmiennych należy pisać czcionką pochyloną, zaś wektory dodatkowo należy pogrubić. Numery wzorów powinny być wyrównane do prawego marginesu. Na przykład:

$$S_i = \frac{1}{V_i} \left[m_i \mathbf{v}_i \otimes \mathbf{v}_i + \frac{1}{2} \sum_{j \neq i}^N \mathbf{r}_{i,j} \otimes \mathbf{f}_{i,j} \right] \quad (1)$$

gdzie m_i jest masą atomu i , \mathbf{v}_i jest jego prędkością, $\mathbf{r}_{i,j} = \mathbf{r}_i - \mathbf{r}_j$ jest wektorem łączącym dwa atomy i oraz j , V_i jest objętością atomu i , $\mathbf{f}_{i,j}$ jest siłą oddziaływań a N określa liczbę atomów.

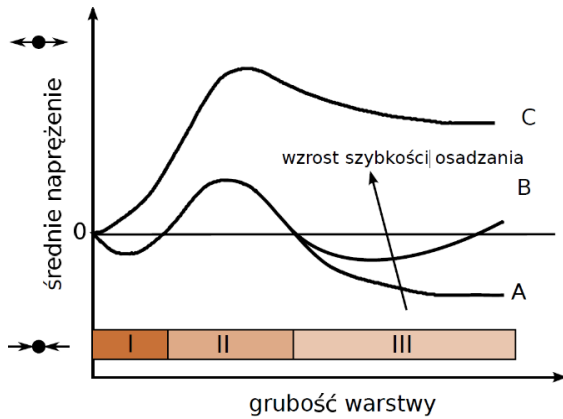
3. Obiekty

Rysunki, tabele, przykłady i algorytmy powinny być umieszczone możliwie blisko pierwszego odwołania się do nich w tekście artykułu. Rysunki, zdjęcia, rzuty ekranu lub ogólnie rozumiana grafika powinna być

dobrej jakości po umieszczeniu w treści artykułu. Powinny być czytelne i osadzone trwale w dokumencie.

3.1. Rysunki

Rysunki powinny być wyśrodkowane względem kolumny a podpisy znajdować się pod rysunkiem, tak jak na Rysunku 1. Możliwe jest umieszczanie rysunku zajmującego całą szerokość strony.



Rysunek 1: Rysunek powinien być czytelny, może być w formacie JPG, PNG, EPS, PDF w rozdzielczości 300 dpi

3.2. Tabele

Tabele powinny być numerowane i tytułowane nad tabelką jak w przykładzie (Tabela 1) oraz wyśrodkowane względem kolumny. Jednostki muszą być podane w oddzielnych liniijkach.

Tabela 1: Odchylenie standardowe pomiaru odległości

Model	std.dev. $d_n(3)$ (m)	std.dev. $d_n(20)$ (m)	avarage cpu time (s)
M1	0,0021	0,0361	0,95
M2	0,0020	0,0377	1,18
M3	0,0013	0,0190	1,38

Tabela 2: Ta sama tabelka, ale inny wygląd

Model	std.dev. $d_n(3)$ (m)	std.dev. $d_n(20)$ (m)	avarage cpu time (s)
M1	0,0021	0,0361	0,95
M2	0,0020	0,0377	1,18
M3	0,0013	0,0190	1,38

Wielkość czcionki użytej w tabeli może być mniejsza, od czcionki użytej dla tekstu, maksymalnie o dwa punkty. Możliwe jest umieszczenie tabeli zajmującej całą szerokość strony.

3.3. Listingi, kody źródłowe

Przykładowe kody programów powinny być umieszczane w postaci tekstowej z zachowaniem, właściwego formatowania lub w postaci dobrej jakości grafiki, z zachowaniem proporcjonalności wielkości użytej czcionki kodu (patrz Listing 1). Dla polepszenia czytelności kodu można użyć innych czcionek poza bazową. Niezależnie od sposobu umieszczenia podpisujemy je na górze, tak jak w przykładowym kodzie.

Listing 1: Przykładowy kod programu

```
#include <iostream>
using namespace std;
void delta(int a, int b, int c)
{
    int delta=b*b-4*a*c;
    cout<<"Delta wynosi: "<<delta;
}
void szescian(int a)
{
    int szescian=a*a*a;
    cout<<"Szescian wynosi: "<<szescian;
}
int main()
{
    delta(5,3,4);
    szescian(3);
    return 0;
}
```

3.4. Algorytmy

Przykładowy Algorytm 1 może wyglądać następująco:

Algorytm 1: Przykład użytego algorytmu

```
input :  $N_s$ -number of experimental points for
        each distance
output: An average path-loss exponent
for  $k = 1$  to  $N$  do
    Generate  $N$  random particles according to
    propagation model
end
for  $j = 1$  to  $N_s$  do // loop over ...
    for  $k = 1$  to  $N$  do
        Calculate  $N$  samples using 1 and
        corresponding weights  $w_k$ 
    end
    Calculate total weight:  $w_s = \sum_{i=1}^N w_k^j$ 
    for  $k = 1$  to  $N$  do
        Normalize weights:  $w_k = w_k / w_s$ 
    end
    Calculate  $N_{eff}$ 
end
Estimate mean path-loss exponent
```

4. Listy uporządkowane i nieuporządkowane

Listy uporządkowane i nieuporządkowane należy przygotować zgodnie ze zdefiniowanymi stylami. Dozwolona jest zmiana stylu numerowania list.

Przykładowa lista uporządkowana:

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.
 - a) Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, lputatea, magna:
 - i. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore.
 - ii. Fugiat nulla pariat. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
 - b) Lorem ipsum dolor sit amet, consectetur.
2. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.
3. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.

Przykładowa lista nieuporządkowana:

- lorem ipsum dolor sit amet, consectetur adipiscing elit Ut purus elit,
- lorem ipsum dolor sit amet, consectetur adipiscing elit:
 - lorem ipsum dolor sit amet, consectetur adipiscing elit,
 - lorem ipsum dolor sit amet, consectetur adipiscing elit,
- lorem ipsum dolor sit amet, consectetur adipiscing elit Ut purus elit.

5. Wnioski

Wnioski powinny zawierać krótkie podsumowanie najciekawszych wyników zawartych w pracy. Ponadto można zamieścić informacje na temat prowadzenia badań w ramach finansowanego programu badawczego; podziękowania dla firm/osób współpracujących przy prowadzeniu badań; informację o zgodzie Komisji ds. Etyki Badań Naukowych w sytuacji, gdy w badaniach przeprowadzono eksperymenty, w których uczestniczyli ludzie.

Literatura

Pozycje w spisie literatury [1-3] powinny być numerowane [4] w nawiasach kwadratowych [3, 5-19] w kolejności występowania w treści artykułu. Nazwa rozdziału **Literatura** nie powinna być numerowana. Kolejne pozycje literaturowe należy umieszczać jak w podanym przykładzie. Preferujemy opisy literatury w języku angielskim (pozycje w spisie literatury). Odniesienia do stron www [3, 4] w spisie literatury oraz do adresów DOI powinny być aktywne i umieszczane tak jak w przykładzie [6-9].

Literatura

- [1] G. Dziewit, J. Korczyński, M. Skublewska-Paszkowska, Performance analysis of relational databases Oracle and MS SQL based on desktop application, *Journal of Computer Sciences Institute* 8 (2018) 263–269, <https://doi.org/10.35784/jcsi.693>.
- [2] W. Bolesta, Analysis of query execution speed in the selected NoSQL databases, *J. Comput. Sci. Inst.* 7 (2018) 138–141, <https://doi.org/10.35784/jcsi.662>.
- [3] Insert a short description of the link. *TEX–basics*, https://en.wikibooks.org/wiki/LaTeX/Plain_TeX, [23.01.2024].
- [4] J. Hsu, C. Fuentes-Hernandez, A.R. Ernst, J.M. Hales, J.W. Perry, B. Kippelen, Linear and nonlinear optical properties of Ag/Au bilayer thin films, *Optics Express* 20 (2012) 8629–8640.
- [5] D. V. Nesterenko, Surface plasmon sensing with different metals in single and double layer configurations, *Appl. Opt.* (2012) 6673–6682, <http://dx.doi.org/10.1007/s11468-013-9575-1>.
- [6] E. Casalicchio, V. Percibali, Measuring Docker Performance: What a mess!!!, *Proceedings of the 8th*

ACM/SPEC on International Conference on Performance Engineering Companion (2017) 11–16, <https://doi.org/10.1145/3053600.3053605>.

- [7] C. Wressnegger, F. Yamaguchi, A. Maier, K. Rieck, Twice the bits, twice the trouble: Vulnerabilities induced by migrating to 64-bit platforms, *Proceedings of the ACM Conference on Computer and Communications Security* (2016) 541–552, <https://doi.org/10.1145/2976749.2978403>.
- [8] J. Stubbs, W. Moreira, R. Dooley, Distributed systems of microservices using Docker and Serfnode, In *7th International Workshop on Science Gateways* (2015) 34–39, <https://doi.org/10.1109/iwsg.2015.16>.
- [9] P. G. de Gennes, *Scaling Concepts in Polymer Physics*, Cornell University Press, London, 1979.
- [10] S. Ambler, *User Interface Design: Tips and Techniques*, Cambridge University Press, Toronto, 2000.
- [11] M. Gajewski, W. Zabierowski, Analysis and comparison of the Spring framework and play framework performance, used to create web applications in Java, In *IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design* (2019) 170–173.
- [12] S. Pavkov, I. Franković, N. Hoić-Božić, Comparison of game engines for serious games, In *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) IEEE* (2017) 728–733.
- [13] N. Ndou, R. Ajoodha, A. Jadhav, Music genre classification: A review of deep-learning and traditional machine learning approaches, In *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRON-ICS)* (2021) 1–6.
- [14] A. M. Potdar, D. G. Narayan, S. Kengond, M. M. Mulla, Performance evaluation of docker container and virtual machine, *Procedia Computer Science* 171 (2020) 1419–1428.
- [15] J. Jagiełło, Performance comparison between React Native and Flutter, Bachelor thesis, Umeå University, Umeå, 2019.
- [16] M. Olsson, A comparison of Performance and Looks Between Flutter and Native Applications: When to prefer Flutter over native in mobile application development, Bachelor thesis, Blekinge Institute of Technology, Karlskrona, 2020.
- [17] M. Jusięga, Comparative analysis of selected versions of the Symfony programming framework, Master thesis, Lublin University of Technology, Lublin, 2020.
- [18] R. Torkar, *Towards Automated Software Testing: Techniques, Classifications and Frameworks*, PhD dissertation, Blekinge Institute of Technology, 2006.
- [19] T. Zientarski, D. Chocyk, Structure and stress in Cu/Au and Fe/Au systems: A molecular dynamics study, *Thin Solid Films* 562 (2014) 347–352.