

2018

Automating of Nanoindentation raw results processing with Python module

Lazarević Jovana, Drljača Mihailo, Brdar Sanja, Petrović Bojan, Kojić Sanja, Stojanović Goran

Lazarević, Jovana, Drljača, Mihailo, Brdar, Sanja, Petrović, Bojan, Kojić, Sanja, et al. 2018. Automating of Nanoindentation raw results processing with Python module. (11–14/06/2018, Palić, Serbia): 1–4. <https://open.uns.ac.rs/handle/123456789/9651> (accessed 29 April 2024). <https://open.uns.ac.rs/handle/123456789/9651>

Downloaded from DSpace-CRIS - University of Novi Sad

Automating of Nanoindentation raw results processing with Python module

Jovana Lazarević, Mihailo Drljača, Sanja Brdar, Bojan Petrović, Sanja Kojić, Goran Stojanović

Abstract—For more than 100 years, researchers in the mechanical sciences have acknowledged that surface contacts between materials are highly reliant on their mechanical properties. In that light, Nanoindentation at small scales has become a conventional tool for the measurement of mechanical properties. Continuing advancements in indentation data analysis increased the method's utility in the characterization of biomaterials. The mechanical characterization of tissues and other biological materials is of utmost importance in clinical medicine and the field of biomaterials. Since the interpretation of nanoindentation raw results appeared as time consuming, the necessity for more rapid approach was recognized. With that in mind the Python module with specific functionality was developed. The enclosed results for multiple test processing with and without proposed optimization show that presented method demonstrates faster and more accurate performance in all cases.

Index Terms— Nanoindentation, biomaterials, Python, data acquisition

I. INTRODUCTION

Nanoindentation testing is a method that determines the mechanical properties of a material in the sub-micron/nano scale. The principal components in a nanoindentation experiment are the sensors and actuators used to apply and measure the mechanical load and indenter displacement, the indenter tip and the test material. Nanoindentation is conducted with a calibrated Berkovich diamond indenter tip. Typically, a specimen is mounted onto a hardened base or specimen mount using a very thin layer of glue [1]. Throughout a nanoindentation test, force, displacement and time on sample are documented as the indenter tip is forced into the test material's surface with a prescribed loading and unloading parameters or defined depth of indentation [2].

Due to heterogeneous texture of bio samples, it is common to launch statistically enough number of tests in order to have more precise insight in their mechanical properties. For presented purpose 50 measurement points per sample are performed. Difficulty that appears is

Jovana Lazarević is with the Faculty of Technical science, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia (e-mail: jovana_lazarevic@uns.ac.rs).

Mihailo Drljača, BioSense Institute, University of Novi Sad, Dr Zorana Đinđića 1, 21000 Novi Sad, Serbia (e-mail: mihailo_drljaca@uns.ac.rs).

Sanja Brdar, BioSense Institute, University of Novi Sad, Dr Zorana Đinđića 1, 21000 Novi Sad, Serbia (e-mail: brdars@uns.ac.rs).

Bojan Petrović is with the Faculty of Medicine, University of Novi Sad, Hajduk Veljkova 3, 21000 Novi Sad, Serbia (e-mail: bojan.petrovic@mf.uns.ac.rs).

Sanja Kojić, is with the Faculty of Technical science, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia (e-mail: sanjakojic@uns.ac.rs).

Goran Stojanović, is with the Faculty of Technical science, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia (e-mail: sgoran@uns.ac.rs).

demanding data acquisition. To be precise, presenting multiple load-displacement curves on one figure requires a disposal of failed and unsuccessful tests. Furthermore, finding average value of each Displacement (Load) per test with a purpose of displaying a global $P-h$ curve of the sample has showed as a time consuming and unpractical to be done manually. The need for automatic processing has been recognized.

The response of interest is the load-displacement curve (often called the $P-h$ curve). Nowadays, the most frequently used nanoindentation model is the Oliver and Pharr model. This model calculates elastic properties of the material from the unloading part of the indentation curve [3].

The analysis of experimentally gained data is based on interpretation of load – displacement curve (Fig. 1). Basic form of the curve consists of a loading to full load followed by an unloading to zero load or with 90% of maximum load on sample [3].

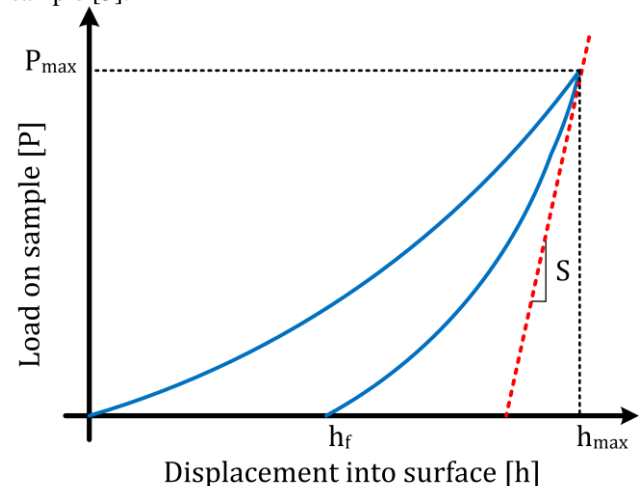


Fig. 1 Representation of a typical load-displacement curve where P_{max} = maximum applied load; h_{max} = penetration depth; h_f = final depth; S = unloading stiffness.

Although nanoindentation was originally developed as a tool for mechanical characterization of films and coatings [4], it has recently developed as a dominant tool for measuring nano and microscale mechanical properties in tissues, bones and other biomaterials. This technique has been used to measure the mechanical properties of microstructural features in bone and teeth, investigate variations in mechanical properties with changes in tissue organization or composition in mineralized and soft tissues, and map mechanical properties spatially in complex biomaterials [5]. However, results of biomaterials nanoindentation are highly influenced by the sample mounting procedure, surface roughness and sample time-dependent behavior [5]. This is especially true for glass ionomer samples due to the difficulty of sample fixation.

Conventional glass-ionomer (GI) cements have a large number of applications in dentistry. One of the advantages

of GI, compared to other restorative materials, is that they can be placed in cavities without any need for bonding agents. As a filling material they demonstrate remarkable biocompatibility [6].

This article describes the implementation of a Python developed calculation algorithm and its application on nanoindentation raw results; calculations have been performed following the requirements of the aforementioned multiple-test processing problem.

II. THE EXPERIMENTAL METHOD

A. Nanoindentation samples and data

Seven glass-ionomer specimens were used as the test samples. This samples were prepared according to the manufacturer's instructions. Immediately before mixing, the materials were activated and then placed in the amalgamator. The trituration process lasted 13 seconds. Immediately after preparation the capsule material was placed in the applicator and the cement was ready for use. The finished material was placed in cylindrical rubber molds with rounded hole, 5mm diameter. On each side of the mold one glass plate was placed, which allowed the creation of a flat surface, which is important for the validity of the measurement. During cementing it, it was under pressure. After curing, a circular cement plate was obtained, a diameter of 5mm, thickness of 2mm serving as a sample. Nanoindentation tests were performed by KeySight Nano Indenter G200 [7]. In further text samples nanoindentation results will be referred as "fuji_I.xls" up to "fuji_VII.xls". All data were extracted in NanoSuite (KeySight Nano Indenter G200 provided software).

III. MODULE CREATION, RESULTS AND DISCUSSION

A. Excel file infrastructure

NanoSuite software generates a separate Excel workbook for each sample. The sample workbook comprises worksheets for the NanoSuite results, required inputs, and editable post-test inputs (Fig. 2). The sample workbook also holds a worksheet for each test, which contains a column for each variable.

The displayed columns are controlled by the channel export order as defined in the NanoSuite method. There are two types of variables: hardware and software. Hardware variables store data attained directly from the instrument. For example, "Time" is a hardware variable. A software variable stores the results of a calculation applied to every acquired data point. For instance, "Load on Sample" and "Displacement into Surface" are software variables. All load and displacement data acquired throughout the test are referenced to the point at which the indenter first contacts the test surface. Any NanoSuite variable can be exported [8].

1	Test	Modulus / Hardness		Drift Corre	Disp at Me	Load At M	Time At S	Tip Name
2		GPa	GPa	nm/s	nm	mN		
3	1	26.532	0.988	0.019	2212.064	97.03	11:53:49	TB15707 08
4	2	28.688	1.224	-0.033	2018.715	97.318	11:59:37	TB15707 08
5	3	35.023	1.692	-0.062	1738.924	96.953	12:07:36	TB15707 08
6	4	****	****	****	****	****	12:16:36	TB15707 08
7	5	31.962	1.228	-0.072	1986.947	96.719	12:20:55	TB15707 08
8	6	25.367	0.944	-0.088	2265.986	97.277	12:30:40	TB15707 08
9	7	22	0.66	-0.232	2652.637	96.86	12:35:48	TB15707 08
10	8	26.246	0.844	-0.131	2362.472	97.15	12:45:41	TB15707 08
11	9	23.956	0.675	-0.108	2613.11	97.072	12:51:04	TB15707 08
12	10	50.262	3.185	-0.114	1319.045	97.167	12:59:06	TB15707 08
13	11	****	****	****	****	****	1:10:21	TB15707 08
14	12	28.688	1.265	-0.161	1986.433	96.713	1:13:32	TB15707 08
15	13	****	****	****	****	****	1:21:48	TB15707 08
16	14	20.197	0.675	-0.155	2645.996	96.742	1:27:22	TB15707 08
17	15	****	****	****	****	****	1:36:17	TB15707 08

Fig. 2. Excel workbook

A "Results" worksheet contains the calculated mean value of Modulus and Hardness at max load as well mean value from each test for all measurements. The first column of this worksheet include serial number of test. Although mean results for every test performed cannot be presented, their total number can be seen from 4th worksheet (in this case "Test 050") (Fig. 3). Several tests in this particular sample failed as evidenced by invalid results ("****" sign). Their worksheet title is marked as "Test number_of_test Tagged".

Test worksheet has the same template for all generated data. Fig. 3. Single test worksheet appearance Fig. 3 shows that each column is a single-dimensional array of size A, where A is the number of data points acquired during a single test. Dimension of A (column length) is not uniform for each test [8].

1	Segment	Displacement	Load On Sample	Time	Stiffness	Modulus	Hardness
2		nm	mN	s	N/m	GPa	GPa
3	Load Segr	0	0	0	1.798E+308	1.798E+308	1.8E+308
4		-0.130938987	0.00015531	0.2	1.798E+308	1.798E+308	1.8E+308
5		0.570478214	0.000220754	0.36	1.798E+308	1.798E+308	1.8E+308
6		0.616005493	0.000275008	0.44	1.798E+308	1.798E+308	1.8E+308
7		0.445075723	0.000345779	0.52	1.798E+308	1.798E+308	1.8E+308
8		99.55634835	1.287656958	0.8	1.798E+308	1.798E+308	1.8E+308
9		186.3307627	2.562733477	1	1.798E+308	1.798E+308	1.8E+308
10		276.3285082	3.831039635	1.2	1.798E+308	1.798E+308	1.8E+308
11		356.4499492	5.106605328	1.4	1.798E+308	1.798E+308	1.8E+308
12		407.3315468	6.397377554	1.6	1.798E+308	1.798E+308	1.8E+308
13		450.6145644	7.708317082	1.8	1.798E+308	1.798E+308	1.8E+308
14		490.5292696	8.986813522	2	1.798E+308	1.798E+308	1.8E+308
15		526.8924575	10.25249342	2.2	1.798E+308	1.798E+308	1.8E+308
16		557.5141527	11.53820484	2.4	1.798E+308	1.798E+308	1.8E+308
17		589.2245057	12.83691284	2.6	1.798E+308	1.798E+308	1.8E+308

Fig. 3. Single test worksheet appearance

B. Python module

The module was created in that way so it could be able to load several files at once and handle them simultaneously. The files have to be entered with full designation and extension (e.g. FileName.xls). Same stands when is requested from user to create new .xls file in order to save processed results. Failed tests ("****") are automatically excluded from further processing. User can also enter the numbers of tests that are going to be additionally excluded (Fig. 4).

Enter names of files separated by comma(','):fuji_I.xls
 Enter the name of the file u would like to save:Processed_results.xls
 Enter numbers of tests in fuji_I.xls you want to exclude separated by comma(','): 2,4,8

Fig. 4. User interface

In the file named by user (in this case Processed_results.xls) two worksheets were generated, “fuji_I average” and “fuji_I”. The module processing outcome will always be two new created worksheets per entered file (template: “FileName average” and “FileName”). The first sheet contains two columns, “Average Displacements” and “Average Loads” that were calculated from remaining tests (Fig. 5). The second sheet contains column couples of “Displacement into surface” and “Load on Sample” of those tests that were included in processing. There is a headline over each couple that indicates the number of test from which they were extracted (Fig. 6).

In order to read, manipulate or write .xls file certain packages were needed including does for data mining (Fig. 7).

Packages are namespaces which contain multiple packages and modules themselves. They are simply directories of Python module(s) [9]. With imported packages following essential functionalities were enabled:

- Data extraction from MS Excel spreadsheet files
- Average value calculation
- Generation of data structure consisted of Displacement-Load pair of previously processed tests
- Creation of spreadsheet files compatible with MS Excel

	A	B
1	Average Displacement	Average Load on sample
2	0	0
3	4.61647533	-0.000195724
4	5.628661244	-0.000156044
5	3.186467588	0.000358418
6	6.382751831	0.000268682
7	-5.793845198	0.001328984
8	6.251128597	0.000581105
9	3.1042984	0.000970829
10	7.961869023	0.000761722
11	5.774486408	0.001078026
12	6.665658562	0.00116538
13	10.52978425	0.00103259
14	6.189704767	0.001512361
15	11.60229471	0.001266388
16	9.018189371	0.001617123
17	10.08403783	0.001695608
18	11.20596931	0.001769434
19	8.454350543	0.002134787
20	15.25538926	0.00178912
21	7.35486302	0.002543831
22	14.53368802	0.0021699
23	12.34387522	0.002498427
24	13.7730008	0.002556222
25	13.4294373	0.002750093
26	18.14108732	0.002565003
27	19.28258989	0.00264539

Fig. 5. Generated average worksheet

	A	B	C	D	E	F
1	0	1	2	3	4	5
2	Test 001	Test 001	Test 003	Test 003	Test 005	Test 005
3	Displacement	Load on	Displacement	Load on	Displacement	Load on
4	0	0	0	0	0	0
5	0.033948813	0.0001	2.597632493	-5E-05	20.00675704	-0.0013
6	-0.17258463	0.0003	4.898763524	-1E-04	22.66215521	-0.0014
7	0.516256712	0.0006	1.827850309	0.0005	6.321802043	0.00011
8	0.358443625	0.0007	5.566203683	0.0003	15.12925556	-0.0004
9	1.08226525	0.0008	-61.5685632	0.0055	17.05141237	-0.0004
10	0.484400346	0.001	9.292571198	0.0004	8.205553363	0.0004
11	1.435124464	0.0011	-15.2995718	0.0024	15.67330585	-1E-05
12	0.843369765	0.0013	14.52105784	0.0003	9.541222645	0.00059
13	1.027053954	0.0014	-6.81126625	0.0021	23.02138003	-0.0003
14	1.34061557	0.0016	6.564775229	0.0012	9.948614423	0.00085
15	1.386020465	0.0017	7.465447455	0.0013	21.37930493	0.00014
16	1.500183133	0.0019	-1.78118258	0.0022	18.22573227	0.00053
17	1.70525866	0.002	20.23010649	0.0007	19.19821379	0.00061
18	1.533689491	0.0022	10.95521216	0.0015	19.39178397	0.00074
19	1.644793075	0.0024	13.25485944	0.0015	18.93747131	0.00093
20	1.888830459	0.0025	10.83346295	0.0019	20.83211766	0.00094
21	2.164190133	0.0026	-7.13934804	0.0034	15.69898291	0.00147
22	2.334884331	0.0028	11.92425064	0.0021	21.09930371	0.00123
23	2.056355302	0.003	12.56361965	0.0022	18.03931957	0.00161
24	2.401997991	0.0031	10.94217709	0.0025	22.56140791	0.00142
25	2.559702852	0.0033	24.15622282	0.0017	23.3684787	0.00152
26	2.527175208	0.0034	14.31899186	0.0026	22.10812878	0.00177
27	2.505343078	0.0036	9.848418016	0.0031	24.90315967	0.00172

Fig. 6. Generated worksheet of all Displacement-Load pair that were included in processing

Package	Version
et-xmlfile	1.0.1
jdcal	1.3
numpy	1.14.1
pandas	0.22.0
pip	9.0.1
python-dateutil	2.6.1
pytz	2018.3
setuptools	28.8.0
six	1.11.0
xlrd	1.1.0
xlutils	2.0.0
xlwt	1.3.0

Fig. 7. List of used packages

Statistically created chart that displays average calculated data from one of the sample results (“fuji_IV.xls”), highlights the outstanding performance and effectiveness of Python module (Fig. 8).

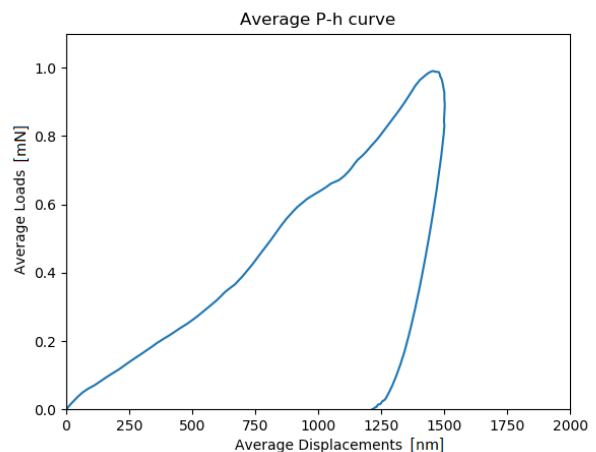


Fig. 8. Average P-h curve

C. Python module performance

In order to show module performances seven Excel files were simultaneously processed. Each of them consisted of fifty tests. Resulting Excel file contained 24 sheets, where 2 sheets per Excel file were generated. Tests that were excluded from further processing were randomly entered. The imported 'timeit' library was used for measuring execution time. Processing time was 1.15 seconds (Fig. 9).

```
Enter names of files separated by comma(','):
fuji_I.xls,fuji_II.xls,fuji_III.xls,fuji_IV.xls,fuji_V.xls,fuji_VI.xls,fuji_VII.xls
Enter the name of the file u would like to save:Processed_results_all.xls
Enter numbers of tests in fuji_I.xls you want to exclude separated by comma(','):
2,4,6,8
Enter numbers of tests in fuji_II.xls you want to exclude separated by comma(','):
3,6,9
Enter numbers of tests in fuji_III.xls you want to exclude separated by comma(','):
1,7
Enter numbers of tests in fuji_IV.xls you want to exclude separated by comma(','):
5,8,10
Enter numbers of tests in fuji_V.xls you want to exclude separated by comma(','):
3,4,6
Enter numbers of tests in fuji_VI.xls you want to exclude separated by comma(','):
7,8,9
Enter numbers of tests in fuji_VII.xls you want to exclude separated by comma(','):
1,5,7
Processing time is 1.15 seconds
```

Fig. 9. Testing time performance of python module

For the sake of comparison, manual processing of raw results was also performed. The assignment that users had to carry out by hand was the same task that python module executes through every iteration. The time users needed to accomplish everything given varied from 3 to 7 days.

IV. CONCLUSION

In this paper an optimization of Nanoindentation raw results processing has been proposed. Results has shown that time needed for numerous Excel file execution can be incomparably shorter when using automatic processing over

manual one. Same stands for handling only one file consisted from 50 to 100 tests. This kind of optimized processing can be applied for similar methods that export raw results in a form of the Excel file.

ACKNOWLEDGMENT

This paper was financed through H2020 RISE project MEDLEM No. 690876 and Serbian national project TR32016.

REFERENCES

- [1] Fischer-Cripps, Anthony C. *Nanoindentation Testing, Nanoindentation*, 2011 pp.21-37, New York : Springer-Verlag , DOI: 10.1007/978-1-4419-9872-9.
- [2] C.A. Schuh, "Nanoindentation studies of materials", *Materials Today*, Volume 9, Issue 5, 2006, pp. 32-40, DOI: [10.1016/S1369-7021\(06\)71495-X](https://doi.org/10.1016/S1369-7021(06)71495-X).
- [3] A.C. Fischer-Cripps, Critical review of analysis and interpretation of nanoindentation test data, *Surface and Coatings Technology*, Volume 200, Issues 14–15, 2006, pp. 4153-4165, DOI: 10.1016/j.surfcoat.2005.03.018.
- [4] Xiaoqin Huang, Assimina A. Pelegri, Mechanical Characterization of Thin Film Materials with Nanoindentation Measurements and FE Analysis, *Journal of Composite Materials*, Vol 40, Issue 15, 2006, pp. 1393 – 1407, DOI: [10.1177/0021998305059728](https://doi.org/10.1177/0021998305059728).
- [5] D. M. Ebenstein, L. A. Pruitt, Nanoindentation of biological materials, *Nano Today*, Volume 1, Issue 3, 2006, Pages 26-33, DOI: /10.1016/S1748-0132(06)70077-9.
- [6] Khoroushi, Maryam, and Fateme Keshani. "A Review of Glass-Ionomers: From Conventional Glass-Ionomer to Bioactive Glass-Ionomer." *Dental Research Journal* 10.4, 2013, pp. 411–420. DOI: 10.1016/B978-1-4557-7862-1.00016-X.
- [7] <https://www.keysight.com/en/pd-1675520-pn-U9820A/agilent-nano-indenter-g200?cc=RS&lc=eng> last visited: 03.13.2018.
- [8] Agilent Technologies, Inc. *G200 Nano Indenter User's Guide*.2009, USA.
- [9] D. Kuhlman, "A Python Book: Beginning Python, Advanced Python, and Python Exercises", Massachusetts, USA, 2009.