

# Wprowadzenie do CalcT<sub>E</sub>X-a – dokument w budowie

Warszawa, 1 czerwca 2008

# Spis treści

<b>1</b>	<b>Wprowadzenie do CalcTeX-a</b>	<b>3</b>
1.1	Co to jest CalcTeX? . . . . .	3
1.2	Wymagania systemowe . . . . .	4
1.3	Pierwsze obliczenia . . . . .	4
1.3.1	Plik TeX-a lub L <sup>A</sup> TeX-a do obliczeń . . . . .	4
1.3.2	Główny skrypt kompilacyjny – plik „go” . . . . .	6
1.3.3	Proste działania matematyczne . . . . .	7
1.3.4	Przypisanie wartości zmiennej oraz wydruk wartości zmiennej . . . . .	8
1.3.5	Cytowania w środowiskach do obliczeń dla CalcTeX-a . . . . .	9
1.3.6	Dodawanie źródeł pythona . . . . .	9
1.3.7	Ogólne przykłady . . . . .	11
<b>2</b>	<b>Testy</b>	<b>12</b>
2.1	Proste testy . . . . .	12
2.2	Zadanie z jednostkami i wykresem . . . . .	14
2.2.1	Praca techniczna . . . . .	14
2.2.2	Praca techniczna – rozwiązanie . . . . .	15
2.2.3	Temperatura wody . . . . .	17
2.2.4	Temperatura wody – Rozwiązanie . . . . .	17
<b>3</b>	<b>Stan obecny i możliwości rozwoju</b>	<b>18</b>
3.1	CalcTeX– wady, cechy i co jeszcze warto zrobić . . . . .	18
3.2	Wskazówki i cechy . . . . .	19
3.3	CalcTeX– zalety . . . . .	19
3.3.1	Obliczenia z jednostkami . . . . .	20
3.4	CalcTeX wersja r3 . . . . .	23
<b>4</b>	<b>Podsumowanie</b>	<b>23</b>
<b>5</b>	<b>Ważniejsze źródła</b>	<b>24</b>
5.1	Skrypt konwertujący <code>tex2py.sh ver. 4.0</code> . . . . .	24
5.2	Funkcja licząca wartość silni – <code>silnia.py</code> . . . . .	28
5.3	Funkcja licząca wartość symbolu Newtona – <code>newtona.py</code> . . . . .	29
5.4	Plik <code>go</code> dla prostego przykładu wykorzystującego symbol Newtona . . . . .	29
5.5	Zadanie z pracą techniczną . . . . .	29
5.6	FAQ . . . . .	31
<b>6</b>	<b>Licencja</b>	<b>32</b>

# Przedmowa

Jest to luźna próba opisanie możliwości systemu do obliczeń przy wykorzystaniu elementów składni  $\text{T}_{\text{E}}\text{X}$ -a i  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -a w sposób „zrozumiały” przez `python`a.

Zdaję sobie sprawę, że ten dokument można było przedstawić w krótszej formie i bardziej sprawnie. Dodatkowo, jestem świadomy z ograniczeń programu i mimo paru pomysłów na ich eliminację oraz dalszy rozwój, ciągle nie mam zbyt wiele wolnego czasu więc szukam chętnych do pomocy...:) z tego względu zdecydowałem się na wypuszczenie wersji  $\text{CalcT}_{\text{E}}\text{X}$ -a i tego dokumentu w takiej formie jaka jest obecnie.

Starałem się, by ten dokument był zrozumiały dla początkujących programistów jak i początkujących użytkowników  $\text{T}_{\text{E}}\text{X}$ -a i choćby z tego względu chciałem przedstawić możliwie najprostsze przykłady.

Jestem w pełni otwarty na sugestie i uwagi.

Życzę udanej pracy w  $\text{CalcT}_{\text{E}}\text{X}$ -u i zachęcam do zapoznania się z licencją (strona 32).

*Sławek*

## Wstęp

Dawno bardzo dawno temu... kiedy zacząłem pracować pod systemami Linuxowymi ...:) brakowało mi prostego programu do obliczeń (typu MathCAD), który jednocześnie generowałby sensowny wydruk. Pod UNIX-owymi/Linuxowymi systemami jest dostępnych wiele programów do obliczeń/języków programowania oraz program  $\text{T}_{\text{E}}\text{X}$ / $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  do składu dokumentów zawierających wyrażenia matematyczne. Wydaje się naturalne, by użyć razem tych programów i to w taki sposób, by zapisane wyrażenia matematyczne w  $\text{T}_{\text{E}}\text{X}$ -u/ $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -u mogły być interpretowane przez program do obliczeń/język programowania. Taki właśnie mechanizm udawało mi się wdrożyć w  $\text{CalcT}_{\text{E}}\text{X}$ -u a mianowicie jeżeli chciałem wykonać obliczenia jak poniżej:<sup>1</sup>

„Dla  $x := 2.3$  wielkość  $y := x^2$  wynosi  $y = 5.29$ ”, wówczas tworzyłem plik „wsadowy” do `python`-a lub `bc` w takiej formie, by po „przepuszczeniu” go przez `python`a lub `bc` otrzymać plik  $\text{T}_{\text{E}}\text{X}$ -a,  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -a, np.:

- plik „wsadowy” do `python`a” („examples/0py/0.py”):

```
print "Dla $"
print "x:=2.3"
x=2.3
print "$ wielko\\s\\c $"
print "y:=x^2"
y=x**2
print "$ wynosi $"
print "y="
print y
print "$"
print "\\end"
```

lub w krótszej formie:

```
print "Dla"
x=2.3
print "$x=",x,"$"
print "wielko\\s\\c $y:=x^2$"
y=x**2
print "wynosi $y=",y,"$"
```

- plik „wsadowy” do `bc`:(„examples/0bc/0.bc”):

---

<sup>1</sup>zwykle liczyłem bardziej skomplikowane przypadki...;) ten krótki przykład wybrałem, by w miarę szybko i jasno zobrazować metodologię

```

scale=8
print "Dla $"
print "x:=2.3"
x=2.3
print "$ wielko\\'s\\'c $"
print "y:=x^2"
y=x^2
print "$ wynosi $"
print "y="
y
print "$"
print "\\end"

```

Następnie mogłem wygenerować plik TeX-a:

- „python”: `python 0.py > 0py.tex`
- „bc”: `bc < 0.bc > 0bc.tex`

i po przekompilowaniu TeX-em (`tex 0py.tex`, `tex 0bc.tex`) otrzymywałem zamierzony efekt.

Oczywiście „ręczne” tworzenie takich plików wsadowych wymaga nieco wyobraźni, było i jest kłopotliwe; dlatego też, tworzyłem skrypty pomagające generować tego typu pliki „wsadowe” do pythona na podstawie składni TeX-a. Obecną wersję skryptu konwertującego (`tex2py.sh`) stworzył w większości Robert, mój wkład był głównie koncepcyjny, oczywiście pozwoliłem sobie dodać parę drobizgów.

Robert jeszcze raz Wielkie Dziękuję za Twój prywatny czas poświęcony na ten projekt.

Jak już jestem przy podziękowaniu to dziękuję również Kasi za wyrozumiałość, kiedy pracowałem w domu nad CalcTeX-em.

Ten dokument został tworzony zarówno dla użytkowników CalcTeX-a jak i dla potencjalnych deweloperów, z tego względu, z punktu widzenia użytkownika może on zawierać nadmierne informacje.

Ja osobiście uważam, że lepiej zaciekawić niż zanudzić więc liczę na zrozumienie oraz w celu sprawniejszego zapoznania się z CalcTeX-em proponuję skupić się na przykładach, których źródła są dostępne w tym dokumencie jak i również w katalog „examples” oraz na: <http://sg.bzip.pl/CalcTeX>, oczywiście znajomość TeX-a lub/i L<sup>A</sup>TeX-a ułatwia pracę.

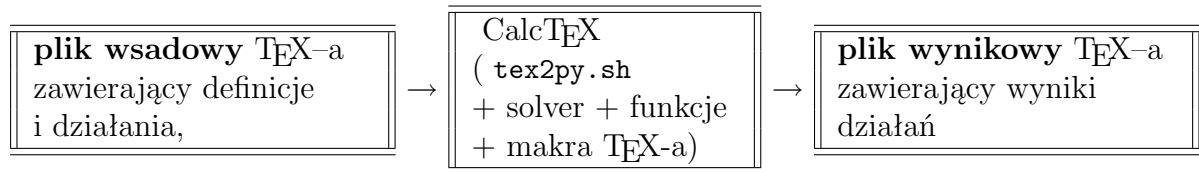
## 1 Wprowadzenie do CalcTeX-a

### 1.1 Co to jest CalcTeX?

CalcTeX jest grupą makr TeX-owych oraz grupą skryptów, w którym najważniejszy jest skrypt `tex2py.sh`, obecnie stworzony przy pomocy „sed”-a, „awk”-a, do interpretacji składni TeX-a w sposób „zrozumiały” przez solver – obecnie pythona. Solver – python generuje pliki TeX-a/L<sup>A</sup>TeX-a wraz z wynikami obliczeń na podstawie odpowiednio sformatowanych plików TeX-a/L<sup>A</sup>TeX-a.

Oczywiście działanie CalcTeX-a można rozszerzyć na inne solwery jak np.: `bc`, `octave`, `maxima`, `SciPy`, itp..

Uwaga w CalcTeX-u obowiązuje zasada predefinicji, czyli zmienna może być użyta pod warunkiem, że jest wcześniej zdefiniowana.



Rysunek 1: Schemat działania CalcTeX-a.

Podsumowując i w najprostszych słowach CalcTeX z pliku TeX-owego ze zdefiniowanymi działaniami tworzy plik TeX-owy z wykonanymi działaniami (rys. 1), dodatkowo nie jest wymagana znajomość programowania w solverze. Z założenia składnia w TeX-owym pliku wsadowym powinna być składnią TeX-a czy L<sup>A</sup>TeX-a tzn., że plik wsadowy powinien być kompilowany przez TeX-a czy też L<sup>A</sup>TeX-a, dodatkowo składnia powinna być niezależna od rodzaju solvera<sup>2</sup>.

## 1.2 Wymagania systemowe

Podstawowe wymagania dla linuxa lub windows<sup>3</sup>: more, sed, awk, latex, python.

Warto też wspomnieć o kodowaniu polskich liter a mianowicie, z racji tego, że testowałem CalcTeX-a na kilku systemach z różnymi stronami kodowymi dla polskich liter, to zdecydowałem się ten dokument i wszystkie przykłady zapisać w notacji „wtyłciachowej”, lub przynajmniej przekonwertowałem dokumenty z dowolnej strony kodowej dla polskich znaków na notację „wtyłciachową”. Dla tej notacji nie miałem żadnych problemów ani z TeX-em ani z pythonem.

## 1.3 Pierwsze obliczenia

Do obliczeń potrzebujemy:

- plik TeX-a lub L<sup>A</sup>TeX-a z odpowiednio zdefiniowanymi działaniami, definicjami;
- skrypt konwertujący ( tex2py.sh) plik TeX-a na składnie zrozumiałą przez pythona.

### 1.3.1 Plik TeX-a lub L<sup>A</sup>TeX-a do obliczeń

Obliczenia, definicje zmiennych oraz wydruk wartości zmiennych są wykonywane przez pythona przy wykorzystaniu pliku ze składnią TeX-a/L<sup>A</sup>TeX-a, ale jedynie pomiędzy znakami:

```

$ ... $
\[ ... \]
\begin{equation}... \end{equation}
  
```

gdzie ... oznaczają: działania, definicje zmiennych, zmienne.

Uwaga: obliczenia powinny być zdefiniowane w jednej linii<sup>4</sup>.

I tak np. jeżeli utworzymy plik TeX-owy o nazwie `ex0ctex.tex` zawierający treść:

<sup>2</sup>jestem świadomy, że nie wszędzie mi się to udało, ale zawsze można to poprawić

<sup>3</sup>testowałem tylko pod cygwinem

<sup>4</sup>można i należałoby to zmienić

W trójkącie prostokątnym mamy: % <----- tu jest tekst  
 $a^2+b^2=c^2$  % <-- tu jest wzór bez obliczeń - cytownie przez Calc\TeX--a  
 wi\c{e}c dla przyprostokątnych % <----- tu jest tekst  
 $a:=3$ , " $b:=4$ ", " $\$$  przeciwprostokątn\c{a} mo\zemy % <-- definicje + tekst  
 obliczy\c z zale\zno\sci: %<----- tu jest tylko tekst  
 $c:=\sqrt{a^2+b^2}$  czyli % <----- definicja zmiennej ,,c'' + tekst  
 d\l ugo\l s\c przeciwprostokątnej wynosi: %< -----tu jest tylko tekst  
 $c$ ." " $\$$  % <----- wyświetlenie obliczonej zmiennej ,,c'' i ,,kropki''  
 \end

jeżeli uruchomimy: `tex2py.sh ex0ctex.tex ex0.tpy`, to wówczas powinniśmy otrzymać plik w formacie pythona: `ex0.tpy` o treści:

```
1 print "W trójkącie prostokątnym mamy:"
2 print "$"
3 print "a^2+b^2=c^2 "
4 print "$"
5 print "wi\c{e}c dla przyprostokątnych"
6 print "$"
7 print "a:=3"
8 a=3
9 print ", "
10 print "$ $"
11 print "b:=4"
12 b=4
13 print ", "
14 print "$ przeciwprostokątn\c{a} mo\zemy"
15 print "obliczy\c z zale\zno\sci:"
16 print "$"
17 print "c:=\sqrt{a^2+b^2}"
18 c=(a**2+b**2)**0.5
19 print "$ czyli"
20 print "d\l ugo\l s\c przeciwprostokątnej wynosi:"
21 print "$"
22 print "c="
23 print c
24 print ". "
25 print "$"
26 print "\end"
```

Po przetworzeniu pythonem:

`python < ex0.tpy > ex0.tex` otrzymamy plik T<sub>E</sub>X-a `ex0.tex`:

```
W trójkącie prostokątnym mamy:          c=
$                                          5.0
a^2+b^2=c^2                               .
$                                          $
wi\c{e}c dla przyprostokątnych          \end
$
a:=3
,
$ $
b:=4
,
$ przeciwprostokątn\c{a} mo\zemy
obliczy\c z zale\zno\sci:
$
c:=\sqrt{a^2+b^2}
$ czyli
d\l ugo\l s\c przeciwprostokątnej wynosi:
$
```

Powyższy plik możemy przekompilować za pomocą T<sub>E</sub>X-a: `pdftex ex0.tex` a otrzymamy plik: `ex0.pdf` dający efekt:

W trójkącie prostokątnym mamy:  $a^2 + b^2 = c^2$  więc dla przyprostokątnych  $a := 3$ ,  $b := 4$ , przeciwprostokątną możemy obliczyć z zależności:  $c := \sqrt{a^2 + b^2}$  czyli długość przeciwprostokątnej wynosi:  $c = 5.0$ .

Wyżej wspomniane działania mogą wydawać się nieco skomplikowane, niemniej jednak można je zamieścić w prostym skrypcie np. o nazwie „go” – dostępny na `examples/ex00/go` o treści:

```
# prosty skrypt do oblicze\n Calc\TeX--em
sh ../../bin/tex2py.sh ex0ctex.tex ex0.tpy # przetworzenie
python < ex0.tpy > ex0.tex #..... obliczenia w pythonie
tex ex0.tex # .....kompilacja \TeX--em
more ex0.tex | grep -v \end >ex0.txt
pdftex ex0.tex # .....kompilacja \TeX--em
```

lub w nieco bardziej zawaansowanej formie jak plik `examples/ex00/go`:

```
echo Glowny skrypt kompilacyjny dla CalcTeX--a
echo S. Golec, 2008-05-25, ver r4
echo przetwarza plik wejsciowy na plik wyjsciowy
echo
InputFile="ex0ctex.tex" # plik wejsciowy
OtputFile="ex0.tex" # plik wyjsciowy
TemporaryFile="a.tpy" # plik pomocniczy

echo usuwanie tymczasowych i pomocniczych plikow
rm *dvi *tpy

echo przetwarzania pliku CalcTeX-a na python-a
sh ../../bin/tex2py.sh $InputFile $TemporaryFile

echo obliczenia w pythonie
python < $TemporaryFile > $OtputFile

echo generowanie pliku z nr. lini w pierwszej kolumnie
more $TemporaryFile | awk 'BEGIN{s=0} {s+=1} {print s"\t"$_}' > ex0-tpy.tex
more $OtputFile | grep -v \end > ex0.txt # wyciecie \end z pliku

echo tex $OtputFile -- kompilacja tex-em
tex $OtputFile; pdftex $OtputFile

echo usuniecie plikow pomocniczych
rm *log *tpy *log *aux
```

Wówczas wystarczy uruchomić plik `go` np.: `sh go`, by wykonać wszystkie w/w kroki. Ten przykład polecam do własnych prób z `CalcTeX-em`.

### 1.3.2 Główny skrypt kompilacyjny – plik „go”

W naszym przykładzie – ten dokument jest kompilowany `LATEX-em` więc dodatkowo wprowadziłem plik `main.tex` zawierający ustawienia dla `LATEX-a`. Ten dokument zawiera kilka części przetwarzanych niezależnie, niemniej jednak struktura pliku kompilacyjnego `go` jest podobna do wcześniej opisanego. (dodatkowo dla lepszej czytelności umieściłem nr lini na początku wiersza):

```
1 echo "
2 echo "S. Golec, 2008-05-25, ver r4.0
3 echo "Główny skrypt kompilacyjny dla CalcTeX--a
4 echo "
5 echo "
6 echo "
7 #sh bin/sh2txt ./bin/prp_tpy_r4a1.sh
8 InputFile="ReadmeCTeX.tex"
9 OtputFile="a.tpy"
10 TemFile="tmp.tpy"
11 more go | awk 'BEGIN{s=0} {s+=1} {print s"\t"$_}' > go.txt # tworzenie pliku „go.txt” z nr linii
12 more bin/tex2py.sh | awk 'BEGIN{s=0} {s+=1} {print s"\t"$_}' > bin/tex2py.sht # jak wyzej
13 #./zmiana < a0.tex > a.ctex # .....zmiana np strony kodowej
14 bin/iso2latex < ReadMeCTeX.tex > ReadMeCTeXpl.tex #....zamiana kodowanie ISO na wty{\l}ciachow{k{a}
15 # przetwarzania
16 #sh bin/tex2pyr4a1.sh PlikWejsciowy plikWyjsciowy
17 sh bin/tex2py.sh ReadMeCTeXpl.tex a.tpy #.....przyk\l{ad 1
18 #sh bin/tex2pyr4d.sh ReadMeCTeXpl.tex a.tpy #.....przyk\l{ad 1
19 # przyk\l{ad obl. energii
20 sh bin/tex2py.sh /examples/energia/ex-energia-ctex.tex /examples/energia/ex-energia-ctex.tpy
```

```

21 sh bin/tex2py.sh unitsc.tex units.tpy #   przetwarzanie definicji jednostek i przedrostk\ow
22 #
23 more units/unitsc.tex           >  examples/pVk/tmp-ctex.tex
24 more examples/pVk/pVk-ctu.tex  >>  examples/pVk/tmp-ctex.tex
25 sh bin/tex2py.sh examples/pVk/tmp-ctex.tex examples/pVk/04-04.tpy #.....przyk\l{}ad 04-04
26 ##
27 sh bin/tex2py.sh examples/solution/solution-calc.tex examples/solution/solution.tpy
28 python < examples/solution/solution.tpy >examples/solution/solution.tex
29 #
30 more bin/pre0.py      > tmp.tpy # .....dodanie czesci definicji do pythona np wlasnych funkcji
31 more bin/silnia.py   >> tmp.tpy # .....dodanie funkcji w pythonie obliczaa\k{a}cej silnie
32 more bin/newton.py  >> tmp.tpy # .....dodanie funkcji w pythonie obliczaj\k{a}cej symbol Newtona
33 #more bin/def.py    >> tmp.tpy # .....wlasnie funkcje pythonowe
34 #more units.tpy    >> tmp.tpy # .....dodanie definicji jednostek i przedrostk\ow
35 more a.tpy         >> tmp.tpy
36 #.....obliczenia w pythonie
37 python < tmp.tpy > a.tex # .....obliczenia w pythonie
38 bin/./tex2iso < a.tex > tmp/ReadMe-CalcTeX-ISO.tex # generowanie dokumentu ze strona kodowa ISO dla PL
39 # generowanie dokumentu ze strona kodowa ISO dla PL
40 #bin/./iso2win2 < tmp/ReadMe-CalcTeX-ISO.tex > tmp/ReadMe-CalcTeX-win.tex
41 # generowanie dokumentu ze strona kodowa ISO dla PL
42 perl bin/iso_win < tmp/ReadMe-CalcTeX-ISO.tex > tmp/ReadMe-CalcTeX-win.tex
43 python < examples/energia/ex-energia-ctex.tpy > examples/energia/ex-energia.tex
44 python < examples/pVk/pVk.tpy > examples/pVk/pVk.tex #....obliczenia w pythonie dla pVk.tpy
45 latex main
46 pdflatex main.tex #.....kompilacja \LaTeX--em
47 latex main.tex ; dvips -o main.ps main.dvi; psbook main.ps | psnup -2 >m2.ps; ps2pdf m2.ps
48 #pdf2ps main.pdf ; psbook main.ps | psnup -2 >m2.ps; ps2pdf m2.ps
49 # latex2html -html_version latin2 html-win; tar -czf main-html.tgz main # tworzenie dok. w formie html
50 # latex2html -init_file bin/html.init main; tar -czf main-html.tgz main # tworzenie dok. w formie html
51 rm -R main/
52 rm *.tpy *2.ps* *.out *log *aux main.ps *toc a.tex *pl.tex go.txt rm *.sht
53 rm *.tpy *out *log main.ps *pl.tex go.txt rm *.sht
54 cp main.pdf Read-Me-CTeX.pdf
55 #tar -czf ../2008-06-03-instructionCalcTeX.tgz ../instruction-r4 # .....tarowanie calego katalogu
56 sh bin/tardocs

```

Należy dodać, żeby ten dokument przekompilować CalcTeX-em, wystarczy rozpakować zawartość <http://sg.bzip.pl/CalcTeX/source> a następnie uruchomić: skrypt go np: `sh go` i jeżeli jest zainstalowany python i TeX to wówczas wszystko powinno zadziałać.

### 1.3.3 Proste działania matematyczne

Ogólnie `tex2py.sh` dla źródła: `$ 1+2 $` generuje działania:

```

print "$"    <-- przepisuje pierwszy znak dolara
print "1+2=" <-- przepisuje dzialania do druku w \TeX-u i dodaje znak ,,='
print 1+2    <-- przepisuje dzialania do obliczen w pythonie
print "$"    <-- przepisuje drugi znak dolara

```

następnie python interperuje działania dając wynik:  $1 + 2 = 3$ .

I tak generalnie, jeżeli plik TeX-a lub L<sup>A</sup>TeX-a przygotowany do obliczeń zawiera źródło to otrzymamy wynik:

źródło	wynik
<code>\$ 1+2 \$</code>	$1 + 2 = 3$
<code>\$ -1+2 \$</code>	$-1 + 2 = 1$
<code>\$ -1+-2 \$</code>	$-1 + -2 = -3$
<code>\$ 2\cdot 3 \$</code>	$2 \cdot 3 = 6$
<code>\$ 2* 3 \$</code>	$2 * 3 = 6$
<code>\$ -2*-3 \$</code>	$-2 * -3 = 6$

```

$ 2^{3} $ 2^3 = 8
$ 2**3 $ ..... 2 * *3 = 8
$ (2^{3})^{4} $ ..... (2^3)^4 = 4096
$ 2^{(3^4)} $ ..... 2^{(3^4)} = 2417851639229258349412352
$ 2^{3\cdot 4} $ ..... 2^{3*4} = 4096
$ 2^{12} $ ..... 2^{12} = 4096
$ 2^{10+2} $ ..... 2^{10+2} = 4096
$ (\pi^{\pi})^{\pi} $ ..... (\pi^\pi)^\pi = 80662.6659386
$ (e^e)^e $ ..... (e^e)^e = 1618.17799191
$ 2\cdot\cdot 3 $ ..... 2 \cdot 3 = 8
$ \frac{3}{2} $ ..... \frac{3}{2} = 1

```

Uwaga! oblicza część całkowitą z dzielenia, natomiast w przypadku „normalnych” obliczeń należy pamiętać o wprowadzeniu liczb „po przecinku”, wówczas wynik będzie poprawny – cecha pythona (łatwo można to zmienić i należałoby to zrobić...).

```

$ \left(2 + \pi\right)/(e+\pi) $ ..... (2 + \pi) / (e + \pi) = 0.87742368362
$ 1.0/2 $ ..... 1.0/2 = 0.5
$ \frac{3.0}{2} $ ..... \frac{3.0}{2} = 1.5
$ \sqrt{2} $ ..... \sqrt{2} = 1.41421356237
$ \sqrt[5]{2} $ ..... \sqrt[5]{2} = 1.148698355
$ \sin(\pi/2) $ ..... \sin(\pi/2) = 1.0
$ \log(e) $ ..... \log(e) = 1.0
$ \text{silnia}\{43\} $ ..... (43)! = 60415263063373835637355132068513997507264512000000000
\l[ 10^{-2}\cdot\sqrt{e^{\pi}}\cdot \sin\left(\frac{2\cdot\pi}{\log\left(e^{\pi}+\pi^e\right)}\right)\r]

```

$$10^2 \cdot \sqrt{e^\pi} \cdot \sin\left(\frac{2 \cdot \pi}{\log(e^\pi + \pi^e)}\right) = 151.949464112$$

$$10^2 = 100; \sqrt{e^\pi} = 1.52367105486;$$

$$(e^\pi + \pi^e) = 45.5998503511; \log(e^\pi + \pi^e) = 3.81990443474$$

$$\frac{2 \cdot \pi}{\log(e^\pi + \pi^e)} = 1.64485405709; \sin\left(\frac{2 \cdot \pi}{\log(e^\pi + \pi^e)}\right) = 0.997258979407.$$

Uwaga:

Co do działań z potęgami należałoby pamiętać o interpretacji dla potęgi potęgi a mianowicie, dla źródła:  $\{2^{\{3\}}\}^4$  dostajemy efekt  $2^{3^4} = 2417851639229258349412352$ , który odpowiada zapisowi  $2^{(3^4)} = 2417851639229258349412352$ . W celu uniknięcia błędów lepiej stosować zapis z nawiasami  $(2^{\{3\}})^{\{4\}}$  co daje efekt:  $(2^3)^4 = 4096$ .

### 1.3.4 Przypisanie wartości zmiennej oraz wydruk wartości zmiennej

Przypisanie wartości zmiennej – definicja zmiennej odbywa się za pomocą znaku „:=” np.:  
źródło efekt

```

$ a:=1.2 $ ..... a := 1.2 .
$ \alpha:=2\cdot \pi $ ..... \alpha := 2 \cdot \pi .

```

Czyli od tej pory pod zmienną  $\alpha$  oraz  $\alpha$  będzie „zapamiętana” jej wartość, ponieważ

<sup>5</sup>funkcja dostępna po dodaniu źródeł funkcji silnia stworzonej w pythonie do CalcTeX-a

`tex2py.sh` „zapamiętuje” zmienne bez wtyłciachów oraz bez nawiasów klamrowych. Jeżeli chcemy wyświetlić zmienną  $\alpha$  wówczas piszemy:

<code>\alpha</code>		efekt
<code>\alpha</code>		$\alpha = 6.28318530718.$
<code>alpha</code>		$alpha = 6.28318530718.$

Na wynikach również można dokonywać działań i to tę właściwość można, na obecnym stanie rozwoju, wykorzystać do obliczeń z jednostkami (patrz późniejszy rozdział 3.3.1 – str. 20):

<code>\alpha/\pi</code>		$\alpha/\pi = 2.0$
<code>\alpha/2</code>		$\alpha/2 = 3.14159265359$
<code>\alpha/(2\cdot\pi)</code>		$\alpha/(2\cdot\pi) = 1.0$

Podobnie zdefiniujemy zmienną  $b$ : `b:=1.2` co daje efekt  $b := 1.2$

wyświetlmy zmienną  $b$  `b` co daje efekt  $b = 1.2$

zdefiniujemy zmienną  $c$  w innym środowisku matematycznym:

<code>\begin{equation}</code>		co daje efekt
<code>c:=\left(\frac{\alpha+b}{b}\right)^{\pi/e}</code>		
<code>\end{equation}</code>		$c := \left(\frac{\alpha + b}{b}\right)^{\pi/e} \quad (1)$

wyświetlimy wartość „ $c$ ” poprzez `c` a otrzymamy  $c = 8.29265776002.$

Przykład z poprzedniego rozdziału:

$$a_1 := 10^2 \cdot \sqrt[e]{\pi} \cdot \sin\left(\frac{2 \cdot \pi}{\log(e^\pi + \pi^e)}\right); \quad a_1 = 151.949464112 \quad (2)$$

### 1.3.5 Cytowania w środowiskach do obliczeń dla CalcTeX-a

Cytowania służą do przepisywania bez żadnej interpretacji części lub całych środowisk matematycznych, oznacza to, że pomiędzy znakami „...” – znakami początku i końca cytowania dla CalcTeX-a, nie wykonywane są obliczenia nawet w środowisku matematycznym. Głównie można używać cytowanie do wprowadzania etykiet w środowiskach matematycznych np: `\label{etykieta}`, do różnego rodzaju strzałek, odstępów: (`\Leftrightarrow`, `\Leftrightarrow`), czy przekształceń wzorów, słowem wszędzie tam, gdzie w środowisku matematycznym nie chcemy wykonywać obliczeń np.:

#### Przykład 03

co daje efekt:

<pre>\begin{equation} c:=b+a " % &lt;-- pocz\k{a}tek cytowania w Calc\TeX--u \;\; \Leftrightarrow \;\; b=\frac{c}{a} \label{cytowanie} ,\;\; \;\; " % &lt;-- koniec cytowania w Calc\TeX--u c % &lt;-- wy\swietlenie warto\`sci zmiennej c \end{equation} w r\`ow. (\ref{cytowanie}) \$c\$</pre>	$c := b + a \Leftrightarrow b = \frac{c}{a}, \quad c = 2.4 \quad (3)$
	w rów. (3) $c = 2.4.$

### 1.3.6 Dodawanie źródeł pythona

W CalcTeX-u istnieje możliwość dodania źródeł pythona i tak np. brakuje funkcji liczącej silnie, można ją dodać, tworząc funkcję w pythonie, należy ją dopisać na po-

czątku pliku przed obliczeniami w pythonie np. jak to jest dostępne dla przykładu: `examples/silnia/go`

```
# S. Golec 2008-05-05
# G{\l}\’owny skrypt zarz\k{a}dzaj\{a}cy dla przyk{\l}adu z~silni\k{a}
# przyk\l{}adowe wykorzystanie dodawania \’zr\’ode\l{} pythona do CalcTeX-a
# poprzez dodanie funkcji liczy\k{a}cej silnie
#‘‘
# przetwarzanie pliku z opisem do pliku do pythona
sh ../../bin/tex2py.sh silnia-calc.tex silnia.tpy
#dodanie bibliotek matematycznych do pythona
more def.py      > a.tmp;          # import bibliotek matematycznych
more silnia.py   >> a.tmp;         # dodanie funkcji silnia
more silnia.tpy >> a.tmp;         # dodanie opisu
cp a.tmp solving.tpy              #
rm a.tmp          #
python < solving.tpy > output.tex # obliczenia w pythonie, wynik jest w output.tex
mv output.tex silnia.tex          #
pdflatex main.tex                 # kompilacja pdflatex-em wynik zapisywany jest do pliku
rm *aux *log *toc *tpy
```

Funkcja licząca silnie w pythonie dostępna jest: `examples/silnia/silnia.py`:

```
#
# 2007-12-03
# funkcja w pythonie - oblicza warto\’s\’c silni
# z liczby naturalnej
#
def silnia(n):
    if n>1:
        return n*silnia(n-1);
    else:
        return 1;
```

po dodaniu definicji `\newcommand{\silnia}[1]{\rm #1 !}` dla TeX-a w pliku np.: „`newcommands.tex`” dla stworzonego źródła otrzymamy efekt:

źródło	efekt
<code>\silnia{2}</code>	$(2)! = 2$
<code>\silnia{3}</code>	$(3)! = 6$
<code>1\cdot 2\cdot 3\cdot 4\cdot 5\cdot 6\cdot 7\cdot 8</code>	$1\cdot 2\cdot 3\cdot 4\cdot 5\cdot 6\cdot 7\cdot 8 = 40320$
<code>\silnia{8}</code>	$(8)! = 40320$
<code>\silnia{6+2}</code>	$(6 + 2)! = 40320$
<code>\silnia{2\cdot 4}</code>	$(2 \cdot 4)! = 40320$
<code>\silnia{2^3}</code>	$(2^3)! = 40320$
<code>\silnia{((\sqrt{8})^2)}</code>	$((\sqrt{8})^2)! = 40320.0$
<code>\silnia{987}/\silnia{986}</code>	$(987)!/(986)! = 987$

Jak mamy kłopoty ze zmieszczeniem wyników np:

```
\silnia{41} $ (41)! = 33452526613163807108170062053440751665152000000000
wówczas można używać „kropki” w zapisie liczby pod silnią np.:
\silnia{(170.0)} $ (170.0)! = 7.25741561531e + 306
```

Uwaga dla liczby powyżej 170.0 zapisanej z „kropką” dostajemy:

$$\$ \backslash\text{silnia}\{171.0\} \$ \quad (171.0)! = \text{inf}$$

ale na takich wynikach można dalej liczyć choć nie zawsze...

$$\$ 171.0/\backslash\text{silnia}\{172.0\} \$ \quad 171.0/(172.0)! = 0.0$$

$$\$ \backslash\text{silnia}\{171.0\}/\backslash\text{silnia}\{172.0\} \$ \quad (171.0)!/(172.0)! = \text{nan}$$

$$\$ \backslash\text{silnia}\{172\}/\backslash\text{silnia}\{171\} \$ \quad (172)!/(171)! = 172$$

Więcej przykładów dotyczących silni można znaleźć na: `examples/silnia`

(`./examples/silnia/main.pdf`). Należałoby przerobić skrypt `tex2py.sh`, by „rozumiał” zapis „3!”. Podobnie możnaby było rozszerzyć działanie skryptu `tex2py.sh`, by „rozumiał” zapis symbolu Newtona.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (4)$$

Przykład obliczania symbolu Newtona z wykorzystaniem funkcji liczącej silnie:

$$\binom{170}{80} = (170)! / ((80)! \cdot (170 - 80.0)!) = 6.8252487219e + 49$$

lub z wykorzystaniem nowej funkcji w pythonie obliczającej symbol Newtona:

$$\binom{170}{80} = \text{Newton}(170, 80.0) = 6.8252487219e + 49.$$

### 1.3.7 Ogólne przykłady

`examples/Ek/ex-Ek-calc.tex`  
`./examples/Ek/main.pdf:`

```
Oblicz Energi\k{e} kinetyczn\k{a} kuli o masie
$m_k:=123.5"\un{kg}"$ poruszaj\k{a}c\k{a}
si\k{e} z\pr\k{e}dko\sci\k{a}
\newcommand{\vv}{\vec{v}}
$\vv_k:=16.0"\un{\tfrac{m}{s}}"$?
\\ Obliczenia:\\ Energi\k{e}
kinetycz\k{a} obliczamy ze wzoru:
\begin{equation}
E_{k}:=\frac{m_k\cdot \vv_k^2}{2}
",\; ; \; \; \; \label{Ek}"
E_k"\un{J}"
```

`\end{equation}`

co daje efekt:

Oblicz Energię kinetyczną kuli o masie  $m_k := 123.5$  kg poruszającą się z prędkością  $\vec{v}_k := 16.0$  m/s ?

Obliczenia:

Energię kinetyczną obliczamy ze wzoru:

$$E_k := \frac{m_k \cdot \vec{v}_k^2}{2}, \quad E_k = 15808.0 \text{ J} \quad (5)$$

`examples/mf/mf-calc.tex:`

```
Woda o masie $m_w:=45.9"\un{kg}"$
przep\l{y}wa w\ci\k{a}gu 1.5 minuty.
Oblicz strumie\`n przep\l{y}waj\k{a}cej masy
wody. \\ Rozwi\k{a}zanie:\\ $sec:=1.0"$, "$
$min:=60.0\cdot sec"$, "$ \\ \%<- def. jednostek
$\tau:=1.5\cdot min$ -- czas przep\l{y}wu
\hfill $\tau"\un{sec}"$
\[\sm_w:=\frac{m_w}{\tau}
",\; ; \; \; \; \%<-- ... tu jest przecinek i spacja
\sm_w \%<-- ..... wydrukowanie warto\`sci
"\un{\frac{kg}{sec}}"$ \%<-- ..... jednostka
\] \%<-- .... koniec \srodowiska matematycznego
```

`examples/mf/mf.pdf` co daje efekt:

Woda o masie  $m_w := 45.9$  kg przepływa w ciągu 1.5 minuty. Oblicz strumień przepływającej masy wody.

Rozwiązanie:

$$sec := 1.0, \quad min := 60.0 \cdot sec,$$

$$\tau := 1.5 \cdot min - \text{czas przepływu } \tau = 90.0 \text{ sec}$$

$$\dot{m}_w := \frac{m_w}{\tau}, \quad \dot{m}_w = 0.51 \frac{\text{kg}}{\text{sec}}$$

W powyższym przykładzie jest nieśmiała próba obliczeń z jednostkami.

## 2 Testy

### 2.1 Proste testy

Plik do testowania... tu jest pole tekstowe może zawierać L<sup>A</sup>T<sub>E</sub>X-owe środowisko matematyczne bez obliczeń np.:  $E = mc^2$  lub poniżej środowisko z obliczeniami:

$2 \cdot 3 = 6$ ,  $(2 \cdot 3)^2 = 36$ ,  $(2^2) \cdot (3^2) = 36$ ,  $(4) \cdot (9) = 36$ .

$$a_1 := 2 \cdot \pi - (\pi^e + e^\pi)^{e/\pi} \cdot \sqrt[e]{e} \cdot \sin\left(\frac{\pi}{2} - 2 \cdot \alpha\right)^{\log(5)/\pi}; \quad a_1 = -31.1854817737 \quad (6)$$

Wielkości  $\pi$  i  $e$  są wcześniej zdefiniowana i wynoszą odpowiednio:  $\pi = 3.14159265359$ ,  $e = 2.71828182846$ .

Zdefiniujmy następujące wielkości:

$$A_{x_1} := \left(\frac{e}{\pi-e}\right)^{e/\pi} \cdot \left(\frac{\pi+e}{e}\right)^{\pi/(e-1)}, \dots, A_{x_1} = 20.3578526188$$

$$a := 1.2345, B_{\beta_a} := 2 \cdot \pi \cdot A_{x_1}^{(\pi-e)^{e/\pi}}, a_p := e/\pi, s_p := 2, \dot{V}_{n_{aaa}} := 5, \dot{m}_a := 5E + 99 \text{ kg/hr}, a_p = 0.865255979432$$

$$b := 2^{3 \cdot \pi} \dots b = 687.291335115$$

$$x_2 := \pi^2 \dots x_2/\pi^2 = 1.0$$

$$\alpha := \frac{\pi}{3}, \quad \beta := \frac{\pi}{3}, \quad (\sin(\alpha))^2 + (\cos(\beta))^2 = 1.0$$

$$\frac{2 \cdot \sqrt{2}}{3} \cdot \cos\left(\frac{\pi}{3}\right) + 4 \cdot \log\left(\frac{e}{5}\right) = -1.96634712895$$

dla

$$\alpha := \frac{\pi}{2}; \quad (\sin(\alpha))^2 + (\cos(\beta))^2 = 1.25.$$

$$\dot{m}_w := a + \left(\frac{e}{\sqrt{\pi}}\right)^{(e/(\pi+\sin(\alpha)))} \quad (7)$$

Wielkość  $\dot{m}_w = 2.55852037701$  z równania (7)...

### Pole koła

Oblicz pole koła, jeżeli jego obwód wynosi:  $O_k := 2 \cdot \sqrt{2} \cdot \pi$ .

Rozwiązanie:

Wiemy, że dla koła:

$$O_k = 2 \cdot \pi \cdot r \Leftrightarrow r := \frac{O_k}{2 \cdot \pi} = \frac{2 \cdot \sqrt{2} \cdot \pi}{2 \cdot \pi} = \sqrt{2}; \quad r = 1.41421356237 \quad (8)$$

więc pole możemy wyliczyć z poniższej zależności:

$$P_k := \pi \cdot r^2 = \pi \cdot \sqrt{2}^2 = 2 \cdot \pi; \quad P_k = 6.28318530718, \quad (9)$$

lub podstawiając:

$$P_{k2} := \pi \cdot \left(\frac{O_k}{2 \cdot \pi}\right)^2 = \frac{O_k^2}{4 \cdot \pi} = \frac{O_k^2}{4 \cdot \pi}; \quad P_{k2} = 6.28318530718, \quad (10)$$

Odp.: Pole koła o obwodzie  $O_k$  wynosi  $O_k^2/(4 \cdot \pi) = 6.28318530718$ .

**Trójkąt prostokątny**

Przyprostokątne mają odpowiednio długości  $a := 3$ ,  $b := 4$  więc długość przeciwprostokątnej można obliczyć:

$$c := \sqrt{a^2 + b^2}, \quad (11)$$

podstawiając do równania (11) mamy:  $c = 5.0$ .

**Energia**

Jaką energię można uzyskać podnosząc ciało o masie  $m_c := 10.0 \cdot 10^{-2}$  kg na wysokość  $\Delta h := 10.0$  m .

**Energia – rozwiązanie**

dla przyspieszenia ziemskiego  $g := 9.81$  m/s<sup>2</sup> mamy:

$$E_p := m_c \cdot g \cdot \Delta h$$

co daje:  $E_p = 9.81$  J.

**Całka**

Oblicz całkę od  $x_1$  do  $x_2$  dla  $x_1 := -e^{\pi/e}/4 \cdot \sin(\pi/3)$  oraz  $x_2 := \pi/3$  z funkcji  $a \cdot x^n$  przy  $a := (e^\pi - \pi^e)^{e/\pi}$  oraz dla  $n \in (1, \infty)$  następnie sprawdź wartość całki dla  $n := 124$ .

$$\int_{x_1}^{x_2} a \cdot x^n dx = \frac{a}{n+1} \cdot x^{n+1} \Big|_{x_1}^{x_2} = \frac{a}{n+1} \cdot (x_2^{n+1} - x_1^{n+1}) = 1.83059311531$$

gdzie:  $x_1 = -0.6876962418$ ,  $x_2 = 1.0471975512$ ,  $a = 0.717669667232$ .

**Inne**

$K := 1000$ ,  $M := 900$ ,  $B := M/10 - 1$ ,

$$K + M + B = 1989 \text{ r.}$$

$1 + 2 - \pi + e = 2.57668917487$

$e + \pi + 10^{67}/1e999 = 5.85987448205$

$1E999 = inf$

$$\pi \cdot \left(\frac{\pi}{e}\right)^{(\sin(\pi/16) \cdot \sqrt{2})} = 3.26957749951$$

$$\S := \left(\sqrt{\pi^{\pi(\cos(\pi))}}\right)^\pi / \left(\sqrt{\pi \sin\left(\frac{\pi}{4}\right)}\right); \quad \S = 8.49347901615$$

$$a_{11} := \frac{1}{2^{2 \cdot 11}}; \quad a_{11} = 2.38418579102e - 07$$

$\frac{1}{2} = 0.5$ ,  $S_8 := \frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} + \frac{1}{2^{2 \cdot 3}} + \frac{1}{2^{2 \cdot 4}} + \frac{1}{2^{2 \cdot 5}} + \frac{1}{2^{2 \cdot 6}} + \frac{1}{2^{2 \cdot 7}} + \frac{1}{2^{2 \cdot 8}}$ ,  $S_8 = 0.83332824707$   
 $\frac{1}{2} + \frac{1}{2^{2 \cdot 1}} = 0.75$ ,  $S_7 := \frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} + \frac{1}{2^{2 \cdot 3}} + \frac{1}{2^{2 \cdot 4}} + \frac{1}{2^{2 \cdot 5}} + \frac{1}{2^{2 \cdot 6}} + \frac{1}{2^{2 \cdot 7}}$ ,  $S_7 = 0.833312988281$   
 $\frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} = 0.8125$ ,  $S_6 := \frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} + \frac{1}{2^{2 \cdot 3}} + \frac{1}{2^{2 \cdot 4}} + \frac{1}{2^{2 \cdot 5}} + \frac{1}{2^{2 \cdot 6}}$ ,  $S_6 = 0.833251953125$

$$\begin{aligned} \frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} + \frac{1}{2^{2 \cdot 3}} &= 0.828125, S_5 := \frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} + \frac{1}{2^{2 \cdot 3}} + \frac{1}{2^{2 \cdot 4}} + \frac{1}{2^{2 \cdot 5}}, S_5 = 0.8330078125 \\ \frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} + \frac{1}{2^{2 \cdot 3}} + \frac{1}{2^{2 \cdot 4}} &= 0.83203125, S_4 := \frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} + \frac{1}{2^{2 \cdot 3}} + \frac{1}{2^{2 \cdot 4}}, S_4 = 0.83203125 \\ \frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} + \frac{1}{2^{2 \cdot 3}} + \frac{1}{2^{2 \cdot 4}} + \frac{1}{2^{2 \cdot 5}} &= 0.8330078125, S_3 := \frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} + \frac{1}{2^{2 \cdot 3}}, S_3 = 0.828125 \\ \frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} + \frac{1}{2^{2 \cdot 3}} + \frac{1}{2^{2 \cdot 4}} + \frac{1}{2^{2 \cdot 5}} + \frac{1}{2^{2 \cdot 6}} &= 0.833251953125, S_2 := \frac{1}{2} + \frac{1}{2^{2 \cdot 1}}, S_2 = 0.75 \\ \frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} + \frac{1}{2^{2 \cdot 3}} + \frac{1}{2^{2 \cdot 4}} + \frac{1}{2^{2 \cdot 5}} + \frac{1}{2^{2 \cdot 6}} + \frac{1}{2^{2 \cdot 7}} &= 0.833312988281 \\ \frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} + \frac{1}{2^{2 \cdot 3}} + \frac{1}{2^{2 \cdot 4}} + \frac{1}{2^{2 \cdot 5}} + \frac{1}{2^{2 \cdot 6}} + \frac{1}{2^{2 \cdot 7}} + \frac{1}{2^{2 \cdot 8}} &= 0.83332824707 \end{aligned}$$

$$\frac{1}{2} + \sum_{i=1}^{10} \frac{1}{2^{2^i}} = \frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} + \frac{1}{2^{2 \cdot 3}} + \frac{1}{2^{2 \cdot 4}} + \frac{1}{2^{2 \cdot 5}} + \frac{1}{2^{2 \cdot 6}} + \frac{1}{2^{2 \cdot 7}} + \frac{1}{2^{2 \cdot 8}} + \frac{1}{2^{2 \cdot 9}} + \frac{1}{2^{2 \cdot 10}} = 0.833333015442$$

$$S_{1 \div 10} := \frac{1}{2} + \frac{1}{2^{2 \cdot 1}} + \frac{1}{2^{2 \cdot 2}} + \frac{1}{2^{2 \cdot 3}} + \frac{1}{2^{2 \cdot 4}} + \frac{1}{2^{2 \cdot 5}} + \frac{1}{2^{2 \cdot 6}} + \frac{1}{2^{2 \cdot 7}} + \frac{1}{2^{2 \cdot 8}} + \frac{1}{2^{2 \cdot 9}} + \frac{1}{2^{2 \cdot 10}}$$

$$S_{11 \div 20} := \frac{1}{2^{2 \cdot 11}} + \frac{1}{2^{2 \cdot 12}} + \frac{1}{2^{2 \cdot 13}} + \frac{1}{2^{2 \cdot 14}} + \frac{1}{2^{2 \cdot 15}} + \frac{1}{2^{2 \cdot 16}} + \frac{1}{2^{2 \cdot 17}} + \frac{1}{2^{2 \cdot 18}} + \frac{1}{2^{2 \cdot 19}} + \frac{1}{2^{2 \cdot 20}}$$

$$S_{21 \div 30} := \frac{1}{2^{2 \cdot 21}} + \frac{1}{2^{2 \cdot 22}} + \frac{1}{2^{2 \cdot 23}} + \frac{1}{2^{2 \cdot 24}} + \frac{1}{2^{2 \cdot 25}} + \frac{1}{2^{2 \cdot 26}} + \frac{1}{2^{2 \cdot 27}} + \frac{1}{2^{2 \cdot 28}} + \frac{1}{2^{2 \cdot 29}} + \frac{1}{2^{2 \cdot 30}}$$

$$S_{1 \div 30} := S_{1 \div 10} + S_{11 \div 20} + S_{21 \div 30}; \quad S_{1 \div 30} = 0.833333333333$$

$$S_{1 \div 10} = 0.833333015442; \quad S_{1 \div 10} + S_{11 \div 20} = 0.833333333333$$

$$\frac{1}{2^{2 \cdot 11}} + \frac{1}{2^{2 \cdot 12}} + \frac{1}{2^{2 \cdot 13}} + \frac{1}{2^{2 \cdot 14}} + \frac{1}{2^{2 \cdot 15}} + \frac{1}{2^{2 \cdot 16}} + \frac{1}{2^{2 \cdot 17}} + \frac{1}{2^{2 \cdot 18}} + \frac{1}{2^{2 \cdot 19}} + \frac{1}{2^{2 \cdot 20}} = 3.17891135637e-07$$

gdzie  $S_{11 \div 20} = 3.17891135637e-07$ .

$$T_1 := 10.0\text{K}, T_2 := 20.0\text{K}, \Delta T_{1 \div 2} := T_2 - T_1, \Delta T_{1 \div 2} = 10.0\text{K}.$$

## 2.2 Zadanie z jednostkami i wykresem

W poniższym przykładzie wykres został stworzony „ręcznie” za pomocą `gnuplot`a i wklejony do zadania, natomiast obliczenia na jednostkach będą omówione w późniejszym rozdziale (roz. 3.3.1, str. 20).

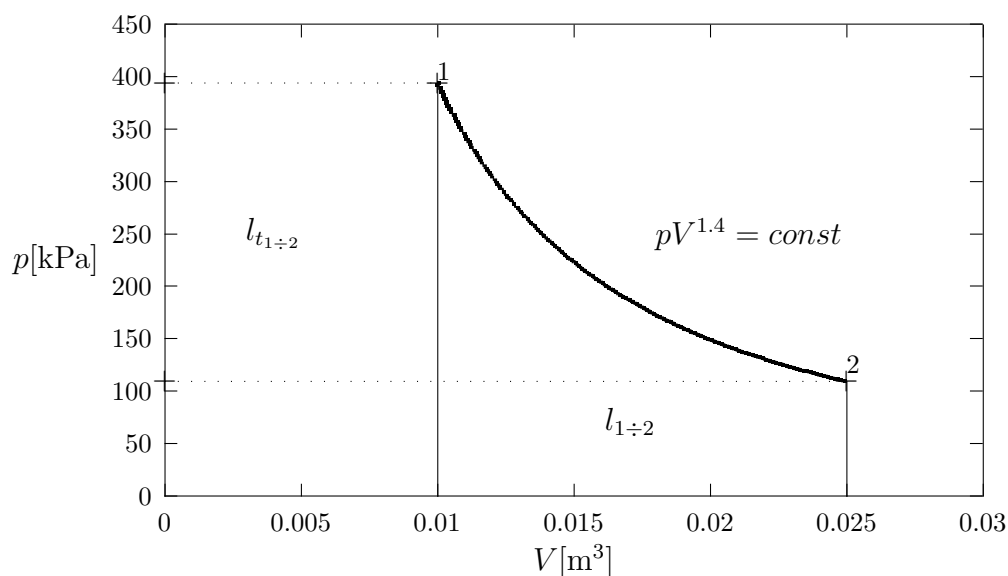
Plik `examples/pVk/pVk-calc.tex ./examples/pVk/pVk.pdf`

### 2.2.1 Praca techniczna

Idealny tłokowy silnik pneumatyczny napełniony jest masą  $m_p := 0.0125 \cdot \text{kg}$  powietrza o objętości  $V_1 := 10 \cdot \text{l}$  przy ciśnieniu manometrycznym  $p_m := 294 \cdot \text{kPa}$ . Podczas ekspansji do objętości  $V_2 := 25 \cdot \text{l}$  ciśnienie zmienia się według zależności:

$$p = p_1 \left( \frac{V_1}{V} \right)^\kappa \quad \kappa := 1.4.$$

Ciśnienie atmosferyczne wynosi  $p_{atm} := 997 \cdot \text{hPa}$ . Jaka jest jednostkowa praca techniczna oddana przez powietrze podczas jednego cyklu roboczego? Jaka jest moc tego silnika, gdy wykonuje on  $f := 5$  cykle robocze w ciągu sekundy?



Rysunek 2: Rozprężanie od  $V_1 = 0.01 \text{ m}^3$  do  $V_2 = 0.025 \text{ m}^3$  w/g krzywej  $pV^\kappa = \text{const}$  dla  $\kappa = 1.4$ .

### 2.2.2 Praca techniczna – rozwiązanie

**Dane:**

$$m_p = 0.0125 \text{ kg}$$

$$V_1 = 0.01 \text{ m}^3$$

$$V_2 = 0.025 \text{ m}^3$$

$$p_m = 294000.0 \text{ Pa}$$

$$\kappa = 1.4$$

$$p_{atm} = 99700.0 \text{ Pa}$$

$$f = 5 \text{ s}^{-1}$$

**Szukane:**

$$l_{t_{1\div 2}} = ?$$

$$N_{1\div 2} = ?$$

Obliczamy ciśnienie absolutne panujące w warunkach początkowych

$$p_1 := p_m + p_{atm}, \quad p_1 = 393700.0 \text{ Pa}$$

$$p_{abs} := p_1 \quad p_{abs} = 393700.0 \text{ Pa}$$

Objętość właściwa

$$v_1 := \frac{V_1}{m_p}; \quad v_1 = 0.8 \frac{\text{m}^3}{\text{kg}}$$

$$v_2 := \frac{V_2}{m_p}; \quad v_2 = 2.0 \frac{\text{m}^3}{\text{kg}}$$

Ciśnienie w punkcie końcowym

$$p_2 := p_1 \cdot \left(\frac{v_1}{v_2}\right)^\kappa; \quad p_2 = 109156.4499 \text{ Pa}$$

Jednoskowa praca techniczna

$$l_{t_{1\div 2}} = - \int_{p_1}^{p_2} v(p) dp \quad (12)$$

wiemy, że

$$\begin{aligned}
 p &= p_1 \left( \frac{v_1}{v} \right)^\kappa \\
 pv^\kappa &= p_1 (v_1)^\kappa \Rightarrow v^\kappa = (v_1)^\kappa \frac{p_1}{p} \\
 v(p) &= v_1 \left( \frac{p_1}{p} \right)^{\frac{1}{\kappa}}
 \end{aligned} \tag{13}$$

Czyli po podstawieniu zależności (13) do równania (12) otrzymamy:

$$\begin{aligned}
 l_{t_1 \div 2} &= - \int_{p_1}^{p_2} v_1 \left( \frac{p_1}{p} \right)^{\frac{1}{\kappa}} dp = -v_1 p_1^{\frac{1}{\kappa}} \int_{p_1}^{p_2} p^{-\frac{1}{\kappa}} dp = -v_1 p_1^{\frac{1}{\kappa}} \left( \frac{\kappa}{\kappa-1} \right) p^{-\frac{1}{\kappa}+1} \Big|_{p_1}^{p_2} \\
 l_{t_1 \div 2} &:= -v_1 \cdot p_1^{(1/\kappa)} \cdot \left( \frac{\kappa}{\kappa-1} \right) \cdot \left( p_2^{((\kappa-1)/\kappa)} - p_1^{((\kappa-1)/\kappa)} \right); \quad l_{t_1 \div 2} = 338264.850699 \frac{\text{J}}{\text{kg}}
 \end{aligned} \tag{14}$$

Obliczenie mocy  $N_{1 \div 2}$

$$N_{1 \div 2} = \frac{L_{t_1 \div 2}}{\tau} = \frac{l_{t_1 \div 2} \cdot m_p}{\tau}; \quad N_{1 \div 2} := l_{t_1 \div 2} \cdot m_p \cdot f; \quad N_{1 \div 2} = 21141.5531687 \text{ W} \tag{15}$$

**Odpowiedź:** Jednostkowa praca techniczna oddana przez powietrze podczas jednego cyklu roboczego wynosi:  $l_{t_1 \div 2} = 338264.850699 \text{ J/kg}$ , natomiast moc tego silnika, gdy wykonuje on  $f = 5$  cykle robocze w ciągu sekundy wynosi:  $N_{1 \div 2} = 21141.5531687 \text{ W}$ .

Plik `examples/solution/solution-calc.tex ./examples/solution/main.pdf`:

## Roztwór gazów

Roztwór gazowy składa się objętościowo z  $r_1 := 40.0/100$  z tlenu ( $M_1 := 32 \text{ kg/kmol}$ ,  $\kappa_1 := 1.4$ ) oraz w pozostałej części z argonu ( $M_2 := 40 \text{ kg/kmol}$ ,  $\kappa_2 := 1.667$ ). Obliczyć: masę molową, stałą gazową, ciepła właściwe  $c_p$ ,  $c_v$ , stosunek ciepła właściwych ( $\kappa_r$ ) dla roztworu.

## Rozwiązanie:

Dane:	Szukane:
$M_1 = 32 \text{ kg/kmol}$ , $M_2 = 40 \text{ kg/kmol}$ ,	$R_r, M_r, \kappa_r, c_{p_r}, c_{v_r}$ .
$\kappa_1 = 1.4$ ,	$\kappa_2 = 1.667$ ,
$r_1 = 0.4$ ,	$r_2 := 1.0 - r_1, r_2 = 0.6$

Obliczam zastępczą masę molową roztworu:

$$M_r := r_1 \cdot M_1 + r_2 \cdot M_2, \quad M_r = 36.8 \text{ kg/kmol} \tag{16}$$

Obliczam stałą gazową roztworu, dla uniwersalnej stałej gazowej  $\tilde{R} := 8314.3 \text{ J/(kmol}\cdot\text{K)}$  :

$$R_r := \frac{\tilde{R}}{M_r} \tag{17}$$

$$\frac{\text{J}/(\text{kmol} \cdot \text{K})}{\text{kg}/\text{kmol}} = \frac{\text{J}}{\text{kmol} \cdot \text{K}} \cdot \frac{\text{kmol}}{\text{kg}} = \frac{\text{J}}{\text{kg} \cdot \text{K}}, \quad R_r = 225.932065217 \frac{\text{J}}{\text{kg} \cdot \text{K}} \quad (18)$$

Obliczam ciepła właściwe roztworu:

$$g_1 := r_1 \cdot \frac{M_1}{M_r}, \quad g_1 = 0.347826086957 \quad (19)$$

$$g_2 := r_2 \cdot \frac{M_2}{M_r}, \quad g_2 = 0.652173913043 \quad (20)$$

$$c_{v_1} := \frac{\tilde{R}}{(\kappa_1 - 1) \cdot M_1}, \quad c_{v_1} = 649.5546875 \frac{\text{J}}{\text{kg} \cdot \text{K}} \quad (21)$$

$$c_{v_2} := \frac{\tilde{R}}{(\kappa_2 - 1) \cdot M_2}, \quad c_{v_2} = 311.630434783 \frac{\text{J}}{\text{kg} \cdot \text{K}} \quad (22)$$

$$c_{v_r} := g_1 \cdot c_{v_1} + g_2 \cdot c_{v_2}, \quad c_{v_r} = 429.169305293 \frac{\text{J}}{\text{kg} \cdot \text{K}} \quad (23)$$

$$c_{p_r} := c_{v_r} + R_r, \quad c_{p_r} = 655.10137051 \frac{\text{J}}{\text{kg} \cdot \text{K}} \quad (24)$$

Obliczam stosunek ciepł właściwych dla roztworu

$$\kappa_r := \frac{c_{p_r}}{c_{v_r}}; \quad \kappa_r = 1.52644041042 \quad (25)$$

Odp: Roztwór charakteryzują następujące stałe: masa molowa:  $M_r = 36.8 \text{ kg}/\text{kmol}$ , stała gazowa:  $R_r = 225.932065217 \text{ J}/(\text{kg} \cdot \text{K})$ , ciepła właściwe:  $c_{p_r} = 655.10137051 \text{ J}/(\text{kg} \cdot \text{K})$ ,  $c_{v_r} = 429.169305293 \text{ J}/(\text{kg} \cdot \text{K})$  oraz stosunek ciepł właściwych:  $\kappa_r = 1.52644041042$ .

### 2.2.3 Temperatura wody

Oblicz o ile wzrośnie temperatura wody po spuszczeniu jej swobodnie z wysokości  $\Delta h_w := 165.5 \text{ m}$  zakładając, że  $\eta_w := 0.65$  część energii uderzenia będzie przekształcona w ciepło oraz pominać opory wody podczas spadania.

### 2.2.4 Temperatura wody – Rozwiązanie

$$\Delta Q = m_w \cdot c_p \cdot \Delta T \Leftrightarrow \Delta T = \frac{\Delta Q}{m_w \cdot c_p} \quad (26)$$

dla

$$\eta \cdot \Delta E_p = \Delta Q \quad (27)$$

oraz  $g_z := 9.81 \text{ m}/\text{s}^2$ ,  $c_p := 4.19 \cdot 10^3 \text{ J}/(\text{kg} \cdot \text{K})$

$$\Delta E_p := m_w \cdot g_z \cdot \Delta h_w \quad (28)$$

po podstawieniu (27) do (26) mamy:

$$\Delta T = \frac{\Delta E_p \cdot \eta}{m_w \cdot c_p} = \frac{(m_w \cdot g_z \cdot \Delta h_w) \cdot \eta_w}{m_w \cdot c_p}, \quad \Delta T := \frac{g_z \cdot \Delta h_w \cdot \eta_w}{c_p}, \quad \Delta T = 0.251864140811^\circ\text{C}$$

Odp.: Jak dla mnie to niewiele się podniesie temperatura, bo o  $\Delta T = 0.251864140811^\circ\text{C}$ , czyli na każdy metr swobodnego spadku przypada przyrost temperatury o  $\Delta T/\Delta h_w = 0.00152183770883 \text{ }^\circ\text{C}/\text{m}$ .

## 3 Stan obecny i możliwości rozwoju

### 3.1 CalcTeX– wady, cechy i co jeszcze warto zrobić

Z pewnością na dzień pierwszy czerwca 2008 roku w CalcTeX-u wymagane jest jeszcze wiele do zrobienia i poprawienia np.:

- dodanie pętli „for”, „while”;
- dodanie struktur „if”;
- z racji używania „sed”-a są problemy z zagnieżdżonymi działaniami: np. wyrażenie zawierające indeksy pod pierwiastkiem w ułamku mogą nie być obliczone np.:

$$\sqrt{\frac{T_1}{T_2} \cdot \frac{T_2 - T_1}{T_2 + 2 \cdot T_2}} \quad (29)$$

jednak powyższe wyrażenie (29) można zapisać w innej – prostszej formie, która będzie już obliczana np.:

$$\left(\frac{T_1}{T_2} \cdot \frac{T_2 - T_1}{T_2 + 2 \cdot T_2}\right)^{0.5} = 0.288675134595$$

lub

$$(T_1/T_2 \cdot (T_2 - T_1) / (T_2 + 2 \cdot T_2))^{0.5} = 0.288675134595;$$

oczywiście z założenia każda poparownie zapisana forma w TeX-u/LaTeX-u powinna być obliczona więc należałoby to poprawić, tymczasowo można spróbować używać prostszej formy zapisu powyższych wyrażen matematycznych, bez używania `\frac{}{}`, `\sqrt{}` i indeksów w zagnieżdżeniu. (błędy wynikają z niepoprawnego działania buforów, wystarczy stworzyć funkcję, która „zapamiętuje” wszystko pomiędzy znakami początku a końca bloku np.: pomiędzy znakami: „{” i „}” czy znakami „\sqrt[” i „]{ ” itp.); oczywiście należy to poprawić;

- dodanie debugera;
- dodanie modułu do obliczeń z jednostkami np.:
 

```
$ F:=2[kN] $ , $ a:=3[km/h] $ dla $ F=m\cdot a, $ $ m:=F/m, $ $ m[kg], $ $ m[t] $
```

 powinien dać efekt:  
 $F := 2[\text{kN}], a := 3[\text{km/h}],$  dla  $F = m \cdot a, m := F/m, m = 2400.0[\text{kg}], m = 2.4[\text{t}],$   
 czyli: automatyczne wyliczanie jednostek (w przykładzie  $m$  w  $[\text{kg}]$ ), automatyczne przeliczanie jednostek (w przykładzie kilogram  $[\text{kg}]$  na tony  $[\text{t}], [\text{kN}], [\text{km}], [\text{h}],$  etc.);
- dodanie interpretarów: `sh`, `pythona`, `maximy`, ... itp. np. poprzez:
 

```
\begin{sh}polecenia pow\l{}oki sh ...\end{sh} ;
```
- dodanie obliczeń symbolicznych – np. przez dodanie solvera `maximy`;
- współpraca z `gnuplotem` – generowanie wykresów (wykresy można generować w `maximie`);

- współpraca z xFig-iem, QCAD-em, METAPOST-em; w celu generacji rysunków w zależności od obliczonych parametrów (rysunki sparametryzowane);
- dodanie możliwości formatu wydruku liczb.
- podział skryptu `tex2py.sh` na moduły.

## 3.2 Wskazówki i cechy

Początkującym użytkownikom CalcTeX-a sugeruję, by przetwarzali dokument z obliczeniami partiami – po paru zmianach w treści lub podzielili obliczenia na mniejsze pliki, ułatwia to wyłapanie błędów składni. Należy pamiętać, że każdy dokument przygotowany do obliczeń w CalcTeX-u powinien być, bez przetwarzania CalcTeX-echem, poprawnie złożony/skompilowany przez TeX-a/LaTeX-a – oczywiście bez wyników działań, w innym przypadku nie będzie on również poprawnie interpretowany przez CalcTeX-a, zazwyczaj nie otrzymam pliku wynikowego lub otrzymany plik będzie przetworzony do miejsca wystąpienia błędu.

Z racji tego, że CalcTeX zapamiętuje zmienne bez nawiasów klamrowych i wtyłciachów \ np.:  $m_{abs} := 1.2$  definiujemy jako `\sm_{abs}` \$ to solver – python będzie używał zmiennej „sm\_abs”. Podobnie jeżeli zdefiniujemy zmienną `\alpha` \$ to solver będzie „pamiętał” zmienną „alpha” (pomija wtyłciach) z tego względu jeżeli napiszemy `\alpha` \$ oraz `alpha` \$ to odpowiednio dostaniemy efekt:  $\alpha = 1.57079632679$  oraz  $alpha = 1.57079632679$ .

Dobrze jest również używać indeksów, aby nie przedefiniować wcześniejszych zmiennych np. masę powietrza należałoby zdefiniować jako: `m_{p}` \$ ( $m_p$ ), a nie tylko jako „m” itp., tym bardziej, że jako „m” może być zdefiniowany metr.

Z założenia składnia CalcTeX-a powinna być niezależna od solvera, niemniej jednak dla pythona nie można używać zmiennej  $\lambda$ , można natomiast stosować  $\lambda$  z indeksem np.:  $\lambda_p := 0.96$ . Oczywiście to można łatwo poprawić.

## 3.3 CalcTeX– zalety

Najlepiej samemu się przekonać o zaletach...), ale najważniejsze to:

1. „Rozumnienie” elementów składni TeX-a/LaTeX-a w sposób umożliwiający obliczenia, co oczywiście umożliwi generowanie profesjonalnych – TeX-owych dokumentów zawierających obliczenia, zarówno w formie książki, raportów, artykułów...itp. czy nawet prezentacji jak to ma miejsce przy użyciu np. klasy `beamer` ... :D. Oczywiście wraz ze wszystkimi zaletami TeX-a/LaTeX-a, który w/g mnie, jest najlepszym programem do składu dokumentów, szczególnie z wyrażeniami matematycznymi.
2. Do ogromnej zalety, na pewno należy również, możliwość używania dowolnych symboli do oznaczenia zmiennych, nawet można używać obrazków jako zmiennych.
3. Możliwość obliczeń pod systemami: UNIX/Linux oraz DOS i Windows.
4. Niewielkie rozmiary plików do obliczeń.

5. Stosunkowo prosty format zapisu plików, praktycznie jest to składnia TeX-a/LaTeX-a.
6. Bazowanie na składni TeX-a/LaTeX-a a więc mamy dostęp do szerokiej gamy narzędzi TeX-owych.

### 3.3.1 Obliczenia z jednostkami

Obecnie CalcTeX nie jest najlepiej przystosowany do obliczeń z jednostkami, oczywiście w przyszłości zamierzałam załączyć moduł przeznaczony do tego typu obliczeń, niemniej jednak, przy wykorzystaniu obecnych możliwości CalcTeX-a można już dokonywać prostych obliczeń z jednostkami.

I tak dla jednostek z układu SI można dokonywać obliczeń na jednostkach podstawowych, jak np.: m (metr), kg (kilogram), Pa (paskal), itp. oraz jednostkach pokrewnych jak np.: mm (milimetr) gm (gram) kPa (kilopaskal) itp. Obecnie, można definiować wielkości w jednostkach pokrewnych bądź podstawowych natomiast wielkości będą „zapamiętywane” w jednostkach podstawowych SI.

W takim wypadku należy zdefiniować dla TeX-a (poprzez newcommand) używane jednostki oraz zdefiniować dla solvera wartości jednostek podstawowych jak i również odpowiednie wartości dla jednostek pokrewnych i tak np:

definicja dla TeX-a:

<code>\newcommand{\meter}{\rm m}</code>	<code>\$ \cm:=10.0^{-2}\cdot\meter \$</code>
<code>\newcommand{\cm}{\rm mm}</code>	co daje efekt:
<code>\newcommand{\mm}{\rm mm}</code>	<code>m := 1.0</code>
<code>\newcommand{\milimeter}{\rm mm}</code>	<code>mm := 10.0<sup>-3</sup> · m</code>
	<code>mm := 10.0<sup>-3</sup> · m</code>
oraz definicja dla solvera:	<code>cm := 10.0<sup>-2</sup> · m</code> <span style="float: right;"><code>cm = 0.01</code></span>
<code>\$ \meter:=1 \$</code>	<code>cm := 10.0<sup>-2</sup> · m</code> <span style="float: right;"><code>cm = 0.01</code></span>
<code>\$ \milimeter:=10.0^{-3}\cdot\meter \$</code>	
<code>\$ \mm:=10.0^{-3}\cdot\meter \$</code>	

wówczas, gdy zdefiniujemy zmienną np. w mm będzie ona „zapamiętana” w jednostkach podstawowych czyli m, np.:

<code>\begin{equation}</code>	co daje efekt:
<code>a:= 154.5\cdot \mm "; \; \; "</code>	
<code>a "\un{m}"</code>	
<code>\end{equation}</code>	<code>a := 154.5 · mm; a = 0.1545 m</code> (30)

<code>\begin{equation}</code>	co daje efekt:
<code>b:=52.5\cdot\cm "; \; \; "</code>	
<code>b "\un{m}"</code>	
<code>\end{equation}</code>	<code>b := 52.5 · cm; b = 0.525 m</code> (31)

$$O := 2 \cdot (a + b); \quad O = 1.359 \text{ m.} \quad (32)$$

Jeżeli wynik chcemy uzyskać w innych jednostkach niż podstawowe, np. w mm wówczas należy zdefiniować „wynikową” (output) jednostkę niepodstawową:

dla T<sub>E</sub>X-a:

```
\newcommand{\omm}{\rm mm}
```

dla solvera:

```
$ \omm:=1.0/\mm $
```

mm := 1.0/mm

i wówczas dla źródła dostaniemy efekt:

źródło

```
$ b\cdot\omm $
```

efekt

$b \cdot \text{mm} = 525.0.$

lub

```
$ b\cdot\omm"\un{mm}" $
```

$b \cdot \text{mm} = 525.0 \text{mm}.$

Jak widać z powyższego przykładu jest jeszcze trochę do zrobienia jeżeli chodzi o jednostki... .

Plik ze zdefiniowanymi jednostkami to plik: `units/unitsc.tex`: o treści:

```

1 %
2 % Ostatnia zmiana: 2008-05-05, SG kompilacja \today
3 %
4 % Definicje przedrostk'ow i jednostek
5 %
6 %\
7 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8 %Definicja przedrostk'ow
9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10 %\
11 \noindent
12 \newcommand{\mikro}{\rm \mu}
13 $\mikro:=10.0^{-4}$
14 \hfill $\mikro$
15 \
16 %-----
17 \newcommand{\mili}{\rm m}
18 $\mili:=10.0^{-3}$
19 \hfill $\mili$
20 \
21 %-----
22 \newcommand{\decy}{\rm d}
23 $\decy:=10.0^{-1}$
24 %\hfill $\decy$
25 \
26 %-----
27 \newcommand{\centy}{\rm c}
28 $\centy:=10.0^{-2}$
29 \hfill $\centy$
30 \
31 \newcommand{\hecto}{\rm h}
32 $\hecto:=10.0^2$
33 \
34 %-----
35 %\hecto$
36 %-----
37 \newcommand{\kilo}{\rm k}
38 $\kilo:=10.0^3$
39 \
40 %-----
41 \newcommand{\mega}{\rm M}
42 $\mega:=10.0^6$
43 \
44 %-----
45 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
46 %Definicja Jednostek \
47 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
48 %-----
49 \newcommand{\meter}{\un{m}}
50 $\meter:=1.0$
51 \
52 \newcommand{\ometer}{\un{m}}
53 $\ometer:=1.0\cdot\meter$ \
54 %\m:=\meter$
55 %-----
56 \newcommand{\dmeter}{\un{dm}}
57 \newcommand{\odmeter}{\un{dm}}
58 \newcommand{\dm}{\un{dm}}
59 \newcommand{\odm}{\un{dm}}
60 $\dmeter:=\decy\cdot\meter$
61 \hfill $\dmeter"\un{m}"$
62 $\dm:=\decy\cdot\meter$
63 \
64 $\odmeter:=1.0/\dmeter$ \
65 $\odm:=1.0/\dmeter$ \
66 %-----
67 \newcommand{\cmeter}{\un{cm}}
68 \newcommand{\ocmeter}{\un{cm}}
69 %\newcommand{\cm}{\un{cm}}
70 \newcommand{\ocm}{\un{cm}}
71 $\cmeter:=\centy\cdot\meter$
72 \hfill $\cmeter"\un{m}"$
73 $\cm:=\centy\cdot\meter$
74 \
75 $\ocmeter:=1.0/\cmeter$ \
76 $\ocm:=1.0/\cmeter$ \
77 %-----
78 \newcommand{\mmeter}{\un{mm}}
79 \newcommand{\ommeter}{\un{mm}}
80 \newcommand{\mm}{\un{mm}}
81 \newcommand{\omm}{\un{mm}}
82 $\mmeter:=\mili\cdot\meter$
83 \hfill $\mili"\un{m}"$
84 $\mm:=\mili\cdot\meter$
85 \
86 $\ommeter:=1.0/\mmeter$ \
87 $\omm:=1.0/\mmeter$ \
88
89 %-----
90 \newcommand{\liter}{\un{l}}
91 $\liter:=\dmeter^3$
92 \hfill $\liter"\un{m}^3"$
93 \
94 \newcommand{\oliter}{\un{l}}

```

```

95  $\oliter:=1.0/\liter$
96  \hfill $\oliter"\un{m^3}"$\
97  %$\oliter"\un{m}"$
98  %-----
99  \newcommand{\kg}{\un{kg}}
100 $\kg:=1.0$ \
101 %-----
102 \newcommand{\pascal}{\un{Pa}}
103 \newcommand{\Pa}{\un{Pa}}
104 \newcommand{\opascal}{\un{Pa}}
105 \newcommand{\oPa}{\un{Pa}}
106 $\pascal:=1.0$ \
107 $\Pa:=1.0$ \
108 %-----
109 \newcommand{\hPa}{\un{hPa}}
110 $\hPa:=\hecto\cdot \pascal$
111 \newcommand{\ohPa}{\un{hPa}}
112 $\ohPa:= 1.0/\hecto \cdot \pascal$
113 \hfill $\hPa"\un{Pa}"$
114 $\ohPa$
115 \
116 %-----
117 \newcommand{\kPa}{\un{kPa}}
118 $\kPa:=\kilo\cdot \pascal$ \
119 \newcommand{\okPa}{\un{kPa}}
120 $\okPa:=1.0/\kilo \cdot \pascal$ \
121 %-----
122 \newcommand{\MPa}{\un{MPa}}
123 $\MPa:=\mega\cdot \pascal$ \
124 %-----

```

Powyższy plik można dokleić jednorazowo na początku przetwarzanego dokumentu, ale widoczne są wówczas wszystkie definicje jednostek, można to pominąć rozbijając ten plik (`units/unitsc.tex`) na część definicji dla TeX-a poprzez:

```
more units/unitsc.tex | grep newcommand > unc.tpy
```

oraz na część definicji dla solvera:

```
more units/unitsc.tex | grep -v print > units.py
```

i wklejając jedynie te części plików. Należy pamiętać, że część zawierającą definicje dla TeX-a można wklejać tylko jeden raz dla przetwarzanego dokumentu.

Powyżej opisane rozbicie jest wykorzystane w przykładzie: `examples/pVk/go`:

```

# S. Golec, 2008-05-25
# Główny plik go dla zadania z praca rozprezania przy wykorzystaniu CalcTeX-a
# http://sg.bzip.pl/CalcTeX/
#-----
#..... Uwaga plik ../../units/unitsc.tex powinien byc w odpowiednim formacie.....
more ../../units/unitsc.tex > units-calc.tex #
#..... przetwarzanie pliku z jednostkami w formacie CalcTeX-a na format pythona
sh ../../bin/tex2py.sh units-calc.tex units-calc.tpy
#..... wyciągnięcie tylko definicji dla pythona - bez wydruku
more units-calc.tpy | grep -v print > units.py
# .....wyciągnięcie definicji dla TeX-a - newcommand
more units-calc.tpy | grep newcommand > unc.tpy
# .....konwertowanie pliku (pVk-calc.tex) w formacie CalcTeX-a na pythona (pVk.tpy)
sh ../../bin/tex2py.sh pVk-calc.tex pVk.tpy
more units.py > tmp.tpy # ..... dodanie definicji jednostek dla pythona
more unc.tpy >> tmp.tpy # ..... dodanie definicji jednostek dla TeX-a
more pVk.tpy >> tmp.tpy # ..... dodanie treści zadania
cp tmp.tpy pVk.tpy # kopiowanie pliku z definicjami jednostek i treścią zadania pVk.tpy
python < pVk.tpy > pVk.tex # ..... generowanie pliku z obliczeniami
latex main # ..... kompilacja latexem
pdflatex main # ..... kompilacja pdflatexem
mv main.pdf pVk.pdf #
rm *.log *.aux *.out tmp* main.dvi units* # ..... usuwanie tymczasowych plików
more pVk-calc.tex | awk 'BEGIN{s=0}{s+=1}{print"s"\t"$_}'> pVk-calc.txt #plik z nr lini

```

Niestety CalcTeX obecnie nie potrafi jeszcze przeliczać jednostek, czyli np.:  $\text{Pa} \cdot \text{m}^2$  nie otrzymamy jeszcze N, ale to zadanie ma wykonywać moduł obliczający jednostki, co wydaje się być łatwiejsze do wykonania używając maximy jako solvera.

### 3.4 CalcTeX wersja r3

Ze względu na potencjalnych deweloperów i ze względu na łatwiejsze zrozumienie działania oraz rozwoju `tex2py` zamieściłem przykład dla wersji r3. Jest to wersja nierozwijana i z licznymi błędami, ale na tej wersji – r3 bazuje wersja r4.

Dla źródła `examples/r3/r3-calc-iso.tex`: Otrzymamy efekt:

```
% plik do obliczen w CalcTeX-u dla ver. r3
Przyk\l{}ad oblicze\n w~Calc\TeX-u w~wer. r3.\
Definicja zmiennych odbywa si\k{e} w~jednej
lini, w~kt\`orej na pocz\k{a}tku jest \# np.: \
$
# a:=2.0
$ -- definicja zmiennej $a$.\
$
# b:=\frac{1}{a}
$. Wy\`swietlenie zmiennej odbywa si\k{e}
po dw\`och haszach, np: $
## a
$, $
## b
$, \begin{equation}
## b^a/\left(a+b\right)\cdot \frac{a-b}{a+b}
\label{eqr3.1}
.\end{equation}
W~r\`ownaniu (\ref{eqr3.1})
wielko\`s\`c: $b^a$ ma warto\`s\`c: $
## b^a
$
```

Przykład obliczeń w CalcTeX-u w wer. r3. Definicja zmiennych odbywa się w jednej linii, w której na początku jest `# np.:`

`a := 2.0` – definicja zmiennej `a`.  
`b :=  $\frac{1}{a}$` . Wyświetlenie zmiennej odbywa się po dwóch haszach, np: `a = 2.0, b = 0.5,`

$$b^a / (a + b) \cdot \frac{a - b}{a + b} = 0.06. \quad (33)$$

W równaniu (33) wielkość: `ba` ma wartość: `ba = 0.25`

Należy dodać, że w wersji r3 nie trzeba używać środowisk matematycznych dla `TeX-a/LaTeX-a` by włączyć tryb do obliczeń więc obliczenia można również wykonywać w plikach tekstowych czy nawet html.

## 4 Podsumowanie

CalcTeX, nawet w obecnym stanie rozwoju, wydaje się bardzo dobrym narzędziem do składu wszelkiego rodzaju dokumentów z obliczeniami a w szczególności zbiorów zadań z rozwiązaniami, projektów inżynierskich, itp.

Ja osobiście używam CalcTeX-a w inżynierskich obliczeniach, gdzie kiedyś używałbym MathCAD-a.

Jest to narzędzie, w którym można budować algorytmy obliczeń przez co po zmianie jednego z parametrów otrzymamy nowy wynik.

Niemniej jednak znajomość składni `TeX-a/LaTeX-a` jest pomocna i przede wszystkim CalcTeX jest dedykowany dla tych użytkowników, którzy choć trochę znają `TeX-a/LaTeX-a`. Dodatkowo znajomość solvera zwiększa możliwości CalcTeX-a, choć dla prostych obliczeń nie jest ona wymagana.

Wydaje się również bardzo kuszące użycie w przyszłości maximy jako solvera, zarówno ze względu na ogromne możliwości maximy np. obliczenia symboliczne, jak i również z powodu generowania przez maximę wyników w postaci `TeX-a`, co mogło by uprościć współpracę solvera z `TeX-em`.

## 5 Ważniejsze źródła

Wszystkie wykorzystane w tym dokumencie źródła są dostępne w katalogu „example”.

### 5.1 Skrypt konwertujący tex2py.sh ver. 4.0

tex2py.sh:

```

1 #!/bin/bash
2 #####
3 # wersja r4.0a1
4 #####
5 # autorzy: Robert Jankowski, Slawomir Golec
6 #####
7 # opis skryptu:
8 #
9 # skrypt do konwersji pliku tekstowego do pythona
10 # - linie z definicjami obliczen powinny zawierac znak :=
11 # - w definicjach dopuszczalne sa spacje
12 # - srodowisko matematyczne definiowane przy uzyciu nastepujacych znacznikow:
13 #   $$
14 #   \[ \]
15 #   \begin{equation} \end{equation}
16 # - tekst pomiedzy znakami " " jest zawsze tylko przepisywany
17 #   (znaczniki ### nie moga wystepowac w tekscie)
18 #
19 #####
20 # start skryptu: tex2py.sh NAZWA_PLIKU_WEJSCIOWEGO NAZWA_PLIKU_WYNIKOWEGO
21 #####
22 # rozniece pomiedzy ver r4
23 # dodanie spoacji przed i po \Delta
24 # poprawa \frac{}{} w definicji
25 #
26 CalcHead="This is file genereted by CalcTeX www.sg.bzip/CalcTeX"
27 ##### przetwarzanie pliku TeX do formatu akceptowalnego przez skrypt awk-sed
28 cat $1 |
29 # usuwanie komentarzy zaczynajacych sie od %
30 sed 's/\%.*//g' |
31 # zamiana znacznikow " na ###
32 sed 's/" / ### /g' |
33 # modyfikacja znacznikow srodowiska matematycznego
34 sed 's/\$/ \$ /g' |
35 sed 's/\\[/ #_begin_eq1 /g' |
36 sed 's/\\]/ #_end_eq1 /g' |
37 sed 's/\\begin{equation}/ #_begin_eq2 /g' |
38 sed 's/\\end{equation}/ #_end_eq2 /g' |
39 # poczatek skryptu awk
40 awk '
41 BEGIN{
42 calc=0;
43 old_calc=0;
44 quote=0;
45 old_quote=0;
46 }
47 {
48 ### tu poczatek skryptu AWK
49
50
51 ### srodowisko matematyczne
52
53 for(i=1;i<=NF;i++) {
54
55 # wykrywanie znacznika srodowiska tekstowego cytowalnego: " "
56 if($i == "###") {
57 if(old_quote==0) quote=1;
58 if(old_quote==1) quote=0;
59 }

```

```
60
61 # wykrywanie znacznika srodowiska matematycznego: $ $
62 if($i == "$" && quote==0) {
63 if(old_calc==0) calc=1;
64 if(old_calc==1) calc=0;
65 }
66
67 # wykrywanie znacznika srodowiska matematycznego: \[ i \]
68 if($i == "#_begin_eq1" && quote==0) calc=1;
69 if($i == "#_end_eq1" && quote==0) calc=0;
70
71 # wykrywanie znacznika srodowiska matematycznego:
72 # \begin{equation} i \end{equation}
73 if($i == "#_begin_eq2" && quote==0) calc=1;
74 if($i == "#_end_eq2" && quote==0) calc=0;
75
76
77 if(calc==0 && old_calc==0) calcs=0
78 if(calc==1 && old_calc==1) calcs=1
79 if(calc==1 && old_calc==0) calcs=2
80 if(calc==0 && old_calc==1) calcs=3
81
82 # w srodowisku tekstowym przepisanie bez zmian
83 if(calcs==0 || quote==1) {
84 if(i==NF) {
85 if(calcs==0) printf("%s\n",$i);
86 if(quote==1 && calcs==1) printf("\n%s\n",$i);
87 }
88 else {
89 if(calcs==0) printf("%s ",$i);
90 if(quote==1 && calcs==1) printf("\n%s ",$i);
91 }
92 }
93
94 # po otwierajacym znaczniku matematycznym przerzut do nowej linii
95 # i wstawienie znacznika #_math
96 if(calcs==2 && quote==0) {
97 printf("%s\n",$i);
98 printf("#_math ");
99 }
100
101 # w kazdej nowej linii w srodowisku matematycznym wstawienie znacznika #_math
102 if(calcs==1 && quote==0){
103 if(i==NF) {
104 printf("%s\n",$i);
105 printf("#_math ");
106 }
107 else
108 printf("%s ",$i);
109 }
110
111 # przed zamykajacym znacznikiem matematycznym przerzut do nowej linii
112 if(calcs==3 && quote==0) {
113 printf("\n");
114 if(i==NF)
115 printf("%s\n",$i);
116 else
117 printf("%s ",$i);
118 }
119
120 # przepisanie znacznika calc na old_calc - zapamietanie starego calc
121 old_calc=calc;
122 # przepisanie znacznika quote na old_quote - zapamietanie starego quote
123 old_quote=quote;
124 }
125 }' |
126 # powtorna zamiana znacznikow srodowiska matematycznego
127 # - powrot do oryginalnych znacznikow
128 sed 's/#_begin_eq1/\\[/g' |
129 sed 's/#_end_eq1/\\]/g' |
```

```

130 sed 's/#_begin_eq2/\\begin{equation}/g' |
131 sed 's/#_end_eq2/\\end{equation}/g' |
132
133 # zamiana znacznika #_math na znaczniki wlasciwe dla typu pola
134 # matematycznego tj. # (obliczenia) lub ## (wyniki)
135 sed '/^#_math $/d' |
136 sed '/^#_math.*:=/ s/#_math/#/g' |
137 sed '/^#_math.*[^\:=]/ s/#_math/##/g' |
138
139 ##### przetwarzanie przygotowanego pliku w formacie posrednim do formatu pythona
140
141 sed '/^#!/ s/\\/\n/g' |
142 # obrobka przy uzyciu awk - nadanie znacznikow liniom roznego
143 # typu: tekst, obliczenia, wydruk obliczen, wyniki, wydruk wynikow
144 awk '{
145 if($1=="#") {
146 # wydruk wzorow matematycznych - jako tekst
147 printf("calc_print %c",34);
148 for(i=2;i<=NF;i++) printf("%s", $i);
149 printf("%c\n",34);
150 # wzory matematyczne do pythona
151 printf("#_calc ");
152 for(i=2;i<=NF;i++) printf("%s", $i);
153 printf("\n");
154 }
155 else {
156 if($1=="##") {
157 # wydruk wynikow obliczen matematycznych - jako tekst
158 printf("res_print %c",34);
159 for(i=2;i<=NF;i++) printf("%s", $i);
160 printf("%c%c\n",61,34);
161 # wyniki matematyczne do pythona
162 printf("##_res ");
163 for(i=2;i<=NF;i++) printf("%s", $i);
164 printf("\n");
165 }
166 else
167 printf("print %c%s%c\n",34,$0,34);
168 }
169 }' |
170
171 # zamiana \left( na ( w wierszach zaczynajacych sie od #_calc
172 # zamiana \right) na ) w wierszach zaczynajacych sie od #_calc
173 # zamiana := na = w wierszach zaczynajacych sie od #_calc
174 # zamiana _{ } na _ w wierszach zaczynajacych sie od #_calc
175 # usuniecie \un{ } w wierszach zaczynajacych sie od #_calc
176 # usuniecie \label{ } w wierszach zaczynajacych sie od #_calc
177 # zamiana \cdot na * w wierszach zaczynajacych sie od #_calc
178 # zamiana ^{ } na **() w wierszach zaczynajacych sie od #_calc
179 # zamiana ^ na ** w wierszach zaczynajacych sie od #_calc
180 # zamiana liczba/liczba na float(liczba)/liczba
181 # sed 's:\([^/][0-9]\{0,\}\)/: float(\1)/:g'
182 sed '/^#_calc/ {
183 s/\\left(/(/g
184 s/\\right)/)/g
185 s/\\un{\([^}]*\)}/g
186 s/\\label{\([^}]*\)}/g
187 s/_{\([^}]*\)}/_ \1/g
188 s/:=/=/g
189 s/\\cdot/*/g
190 s/^\{([^}]*\)}/**(\1)/g
191 #
192 # s:\([^/][0-9]\{0,\}\)/: float(\1)/:g # #/#
193 # s:=: =:g
194 # s:\+: \+:g # dodanie spacji przed + dla #/#
195 # s:[^0-9]\([^/][0-9]\{0,\}\)/: float(\1)/:g #
196 #
197 s/^\**/g }' |
198
199 ### obrobka wierszy zaczynajacych sie od #_calc i zawierajacych \frac

```

```

200 # usuniecie nawiasow { i } z definicji indeksu _{***}
201 # zamiana \frac{ }{ } na ( )/( ) - uzycie buforow
202 sed '/^#_calc.*frac/ {
203 s/_{\([^\)]*\)}\_1/g
204 s/\frac{\([^\)]*\)}{\([^\)]*\)}/(\1)\(/2/g }' |
205 # s/\frac{\([^\)]*\)}{\([^\)]*\)}/float(\1)\(/2/g }' | # chodzi ';'
206 # s/\frac{\([^\)]*\)}{\(.*)}/(\1)\(/2/g }' | # cos bruzdzi bez float
207
208
209 ### obrobka wierszy zaczynajacych sie od #_calc i zawierajacych \sqrt
210 # zamiana pierwiastka kwadratowego \sqrt{ } na ()**0.5 dla wierszy
211 # zaczynajacych sie od #_calc i zawierajacych sqrt{ }
212 sed '/^#_calc.*sqrt{.*/ {
213 s/sqrt{\([^\)]*\)}/(\1)\**0.5/g
214 }' |
215 # zamiana pierwiastka dowolnego stopnia \sqrt[a]{b} na (b)**(1.0/a)
216 # dla wierszy zaczynajacych sie od #_calc i zawierajacych sqrt{ }
217 sed '/^#_calc.*sqrt\[ / {
218 s/sqrt\[ \([^\)]*\) \] \([^\)]*\) /(\2)\**(\1.0\1)/g
219 }' |
220
221
222 ### obrobka wierszy zaczynajacych sie od #_calc
223 # zamiana nawiasow _{ } na _ w wierszach zaczynajacych sie od #_calc - uzycie bufora
224 # zamiana } na )
225 # zamiana } na )
226 # usuniecie \ w wierszach zaczynajacych sie od calc
227 sed '/^#_calc/ {
228 s/_{\([^\)]*\)}\_1/g
229 s/}/)/g
230 s/{{/g
231 s/\\//g }' |
232
233 ### obrobka wierszy zaczynajacych sie od ##_res i zawierajacych \sqrt
234 # zamiana pierwiastka kwadratowego \sqrt{ } na ()**0.5 dla wierszy
235 # zaczynajacych sie od ^##_res i zawierajacych sqrt{ }
236 # zamiana nawiasow _{ } na _ w wierszach zaczynajacych sie od #_calc - uzycie bufora
237 sed '/^##_res.*sqrt{.*/ {
238 s/_{\([^\)]*\)}\_1/g
239 s/sqrt{\([^\)]*\)}/(\1)\**0.5/g
240 }' |
241 # zamiana pierwiastka dowolnego stopnia \sqrt[a]{b} na (b)**(1.0/a)
242 # dla wierszy zaczynajacych sie od ^##_res i zawierajacych sqrt{ }
243 sed '/^##_res.*sqrt\[ / {
244 s/_{\([^\)]*\)}\_1/g
245 s/sqrt\[ \([^\)]*\) \] \([^\)]*\) /(\2)\**(\1.0\1)/g
246 }' |
247
248 ### obrobka wierszy zaczynajacych sie od ##_res i zawierajacych \frac
249 # usuniecie nawiasow { i } z definicji indeksu _{***}
250 # z linii zawierajacej frac - uzycie bufora
251 # zamiana \frac{ }{ } na ( )/( ) - uzycie buforow
252 sed '/^##_res.*frac/ {
253 s/_{\([^\)]*\)}\_1/g
254 s/\frac{\([^\)]*\)}{\([^\)]*\)}/(\1)\(/2/g }' |
255
256 ### obrobka wierszy zaczynajacych sie od ##_res
257 # usuniecie nawiasow { i } oraz \ w wierszach z wydrukiem
258 # wynikow - zaczynajacych sie od #_res
259 #
260 # zamiana \left( na ( w wierszach zaczynajacych sie od ##_res
261 # zamiana \right) na ) w wierszach zaczynajacych sie od ##_res
262 # usuniecie \un{ } w wierszach zaczynajacych sie od ##_res
263 # usuniecie \label{ } w wierszach zaczynajacych sie od ##_res
264 # zamiana _{ } na _ w wierszach zaczynajacych sie od ##_res
265 # zamiana \cdot na * w wierszach zaczynajacych sie od ##_res
266 # zamiana ^{ } na *( ) w wierszach zaczynajacych sie od #_calc
267 # zamiana ^ na ** w wierszach zaczynajacych sie od ##_res
268 # zamiana liczba/liczba na float(liczba)/liczba
269 sed '/^##_res/ {

```

```

270 s/\\left(/(/g
271 s/\\right)/)/g
272 s/\\un{\\([~]*\\)}//g
273 s/\\label{\\([~]*\\)}//g
274 s/_{\\([~]*\\)}_\\1/g
275 s/^{\\([~]*\\)}/**\\1/g
276 s/~/**/g
277 s{/}/g
278 s/}/g
279 s/\\cdot*/g
280 # s:\\+ : \\+:g # dodanie spacji przed + dla #/#
281 # s:[^0-9]\\([^/][0-9]\\{0,\\}\\): float\\1)/:g #
282 s/\\/g }' |
283
284 ### obrobka wierszy zaczynajacych sie od calc_print
285 # dodanie spacji przed i za \\cdot \\Delta
286 # zamiana \\ na \\
287 sed '/^calc_print/ {
288 s/\\cdot/ \\cdot /g
289 s/\\Delta/ \\Delta /g
290 s/\\/\\/\\/g }' |
291
292 ### obrobka wierszy zaczynajacych sie od res_print
293 # dodanie spacji przed i za \\cdot \\Delta
294 # zamiana \\ na \\
295 sed '/^res_print/ {
296 s/\\un{\\([~]*\\)}=\\un{\\1}/g # poprawic by dzialalo a\\un{mm}
297 # s/\\un{\\([~]*\\)}=\\un{\\cdot\\1}/g
298 s/\\cdot/ \\cdot /g
299 s/\\Delta/ \\Delta /g
300 s/\\/\\/\\/g }' |
301
302 sed '/^res_print.*\\un/ {
303 s/=//g }' |
304
305 ### usuniecie znacznikow ###
306 sed 's/###/g' |
307
308 ### usuniecie pustych wydrukow
309 sed '/^print " "$/d' |
310 sed '/^print ""$/d' |
311
312 ### usuwanie pomocniczych znacznikow typu linii
313 # usuniecie znacznika ##_res
314 sed 's/##_res/print/g' |
315 # usuniecie znacznika #_calc
316 sed 's/#_calc //g' |
317 # zamiana znacznika calc_print na print
318 sed 's/calc_print/print/g' |
319 # zamiana znacznika res_print na print
320 sed 's/res_print/print/g' > $2

```

## 5.2 Funkcja licząca wartość silni – silnia.py

bin/silnia.py

```

#
# 2007-12-03
# funkcja w pythonie - oblicza warto\\'s\\'c silni
# z liczby naturalnej
#
def silnia(n):
    if n>1:
        return n*silnia(n-1);
    else:

```

```
return 1;
```

### 5.3 Funkcja licząca wartość symbolu Newtona – newtona.py

```
bin/newton.py
```

```
#
# 2007-12-03
# funkcja w pythonie - oblicza warto\`s\`c symbolu Newtona
#
def Newton(n, k):
    if n>1:
        return silnia(n)/(silnia(k)*silnia(n-k));
    else:
        return 1;
```

### 5.4 Plik go dla prostego przykładu wykorzystującego symbol Newtona

```
examples/silnia/go-newton
```

```
# 2008-05-05
# przyk\l{adowe wykorzystanie dodawania \`zr\`ode\l{}
# pythona do CalcTeX-a poprzez dodanie
# funkcji liczy\k{a}cej silnie i symbol Newtona
#
# przetwarzanie pliku z opisem do pliku do pythona
sh ../../bin/tex2py.sh silnia-ctex.tex silnia.tpy
sh ../../bin/tex2py.sh newton-calc.tex newton.tpy
#dodanie bibliotek matematycznych do pythona
more def.py > a.tmp; # import bibliotek matematycznych
more silnia.py >> a.tmp; # dodanie funkcji silnia
more newton.py >> a.tmp; # dodanie funkcji newton
more silnia.tpy >> a.tmp; # dodanie opisu dla silni
more newton.tpy >> a.tmp; # dodanie opisu dla symbolu Newtona
cp a.tmp solving.tpy
#rm a.tmp
python < solving.tpy > output.tex # obliczenia w pythonie, wynik jest w output.tex
mv output.tex silnia.tex #
pdflatex main.tex # kompilacja pdflatex-em wynik zapisywany jest do pliku main.pdf
```

### 5.5 Zadanie z pracą techniczną

```
examples/pVk/pVk-ctex.tex
```

```
1 %\noindent Ostatnia zmiana: 2008-05-03, kompilacja \today
2 %-----
3 \subsubsection{Praca techniczna}
4 \label{pVk}
5 Idealny t\l{okowy silnik pneumatyczny nape\l{ni}ony jest mas\k{a} $m_p=0.0125\cdot\text{kg}$
6 powietrza o\`obj\k{e}to\`sci $V_1=10\cdot\text{liter}$ przy ci\`snieniu manometrycznym
7 $p_{m}=294\cdot\text{kPa}$. "$ Podczas ekspansji do obj\k{e}to\`sci $V_2=25\cdot\text{liter}$
8 ci\`snienie zmienia si\k{e} wed\l{ug zale\`zno\`sci:
9 \[
10 "p=p_1\left(\frac{V_1}{V}\right)^{\kappa}
11 \];\];\];
12 "
13 \kappa=1.4
14 ". "
```









