CICM 2016, DML Track

A Smooth Transition to Modern mathoid-based Math Rendering in Wikipedia with Automatic Visual Regression Testing

Moritz Schubotz & Alan Sexton
(with material from Frédéric Wang)

Overview

1. Introduction to Wikipedia's new math rendering using mathoid
2. Improvements over the old math rendering using texvc
3. Comparing different math rendering engines using mathpipe
    1. Automation Framework
    2. Comparison of math images
4. Future Work

Overview

1. **Introduction to Wikipedia's new math rendering using mathoid**
2. Improvements over the old math rendering using texvc
3. Comparing different math rendering engines using mathpipe
    1. Automation Framework
    2. Comparison of math images
4. Future Work

Bell's theorem

From Wikipedia, the free encyclopedia
(Redirected from Bell inequality)

Bell's theorem is a 'no-go theorem' that draws an important distinction between

Quantum mechanics

$$\hat{H}|\psi(t)\rangle = i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle$$

Schrödinger equation

Introduction · Glossary · History

Background [show]

Fundamentals [hide]

Complementarity · Decoherence ·

The Free Encyclopedia

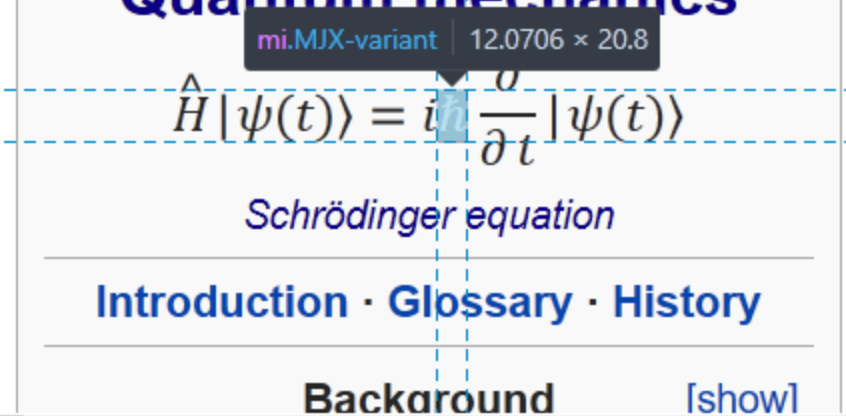
From Wikipedia, the free encyclopedia

(Redirected from Bell inequality)

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Bell's theorem is a 'no-go theorem' that draws an important

Quantum mechanics



Inspekt... Konso... Debugg... Stilbearbeit... Laufzeitana... Netzwerka...

...line.mwe-math-mat... > math > semantics > mrow.MJX-TeXAtom-ORD > mstyle > mi.MJX-variant

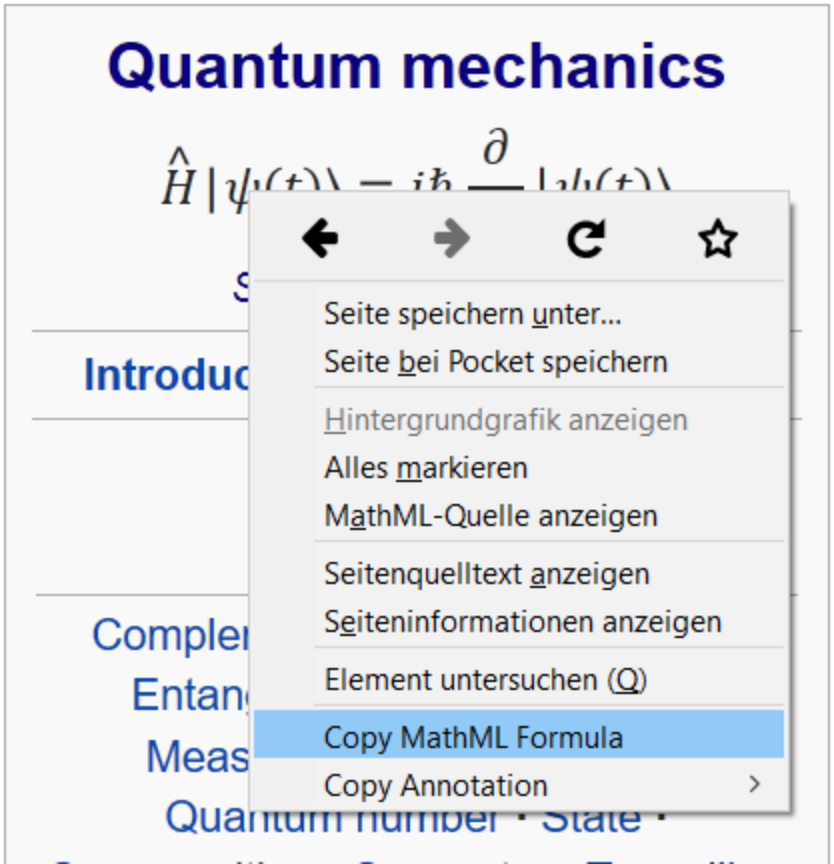
```
<mstyle displaystyle="true" scriptlevel="0">
  <mrow class="MJX-TeXAtom-ORD"></mrow>
  <mrow class="MJX-TeXAtom-ORD"></mrow>
  <mi></mi>
  <mo stretchy="false">(</mo>
  <mi>t</mi>
  <mo stretchy="false">)</mo>
  <mo fence="false" stretchy="false"></mo>
  <mo>=</mo>
  <mi>i</mi>
  <mi class="MJX-variant">
    ℏ
    <!--ℏ-->
  </mi>
```

Bell's theorem

From Wikipedia, the free encyclopedia

(Redirected from Bell inequality)

Bell's theorem is a 'no-go theorem' that draws an important distinction between quantum mechanics (QM) and the



$$\hat{H}|\psi(t)\rangle = i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle$$

(Strg)

$$p = \frac{h}{\lambda}$$

In[8]:=
$$p == \frac{h}{\lambda}$$

Out[8]=
$$p == \frac{h}{\lambda}$$

De Broglie übertrug dies 1923 auf beliebige Teilchen:

$$\lambda = \frac{h}{p}$$

In[9]:=
$$p = \frac{m\,v}{\sqrt{1 - \frac{v^2}{c^2}}}$$

mit dem relativistischen Impuls:

$$p = \frac{mv}{\sqrt{1 - \frac{v^2}{c^2}}}$$

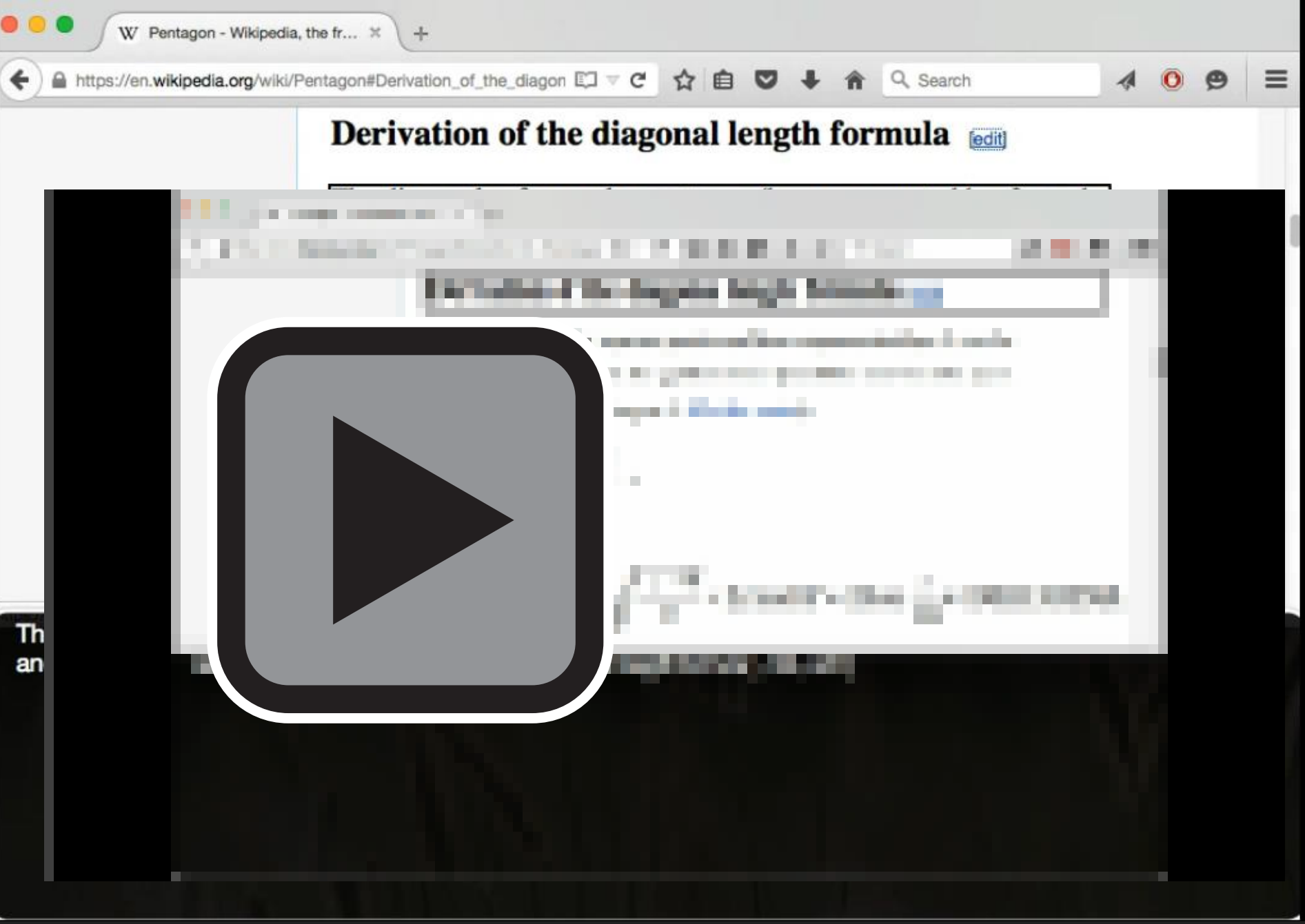
Out[9]=
$$\frac{m\,v}{\sqrt{1 - \frac{v^2}{c^2}}}$$

ergibt sich daraus die sogenannte De-Broglie-Wellenlänge:

In[10]:=
```
Solve[%8, λ]
```

Out[10]=
$$\left\{\left\{\lambda \to \frac{h\sqrt{\frac{c^2 - v^2}{c^2}}}{m\,v}\right\}\right\}$$

$$\lambda = \frac{h \cdot \sqrt{1 - \frac{v^2}{c^2}}}{mv}$$


Pentagon - Wikipedia, the fr...
https://en.wikipedia.org/wiki/Pentagon#Derivation_of_the_diagon
Search
Derivation of the diagonal length formula
edit

Presentation MathML

- Enables Copy and Paste
- Good presentation
- Accessibility
- Better text integration
- Reduces bandwidth
  - Wikimedia 15 Billion visits
  - 5 pages per **person** and month (p/**p**m)
  - Central African Republic 0.1 p/**p**m
  - Ireland 19p/**p**m

https://addons.mozilla.org/firefox/addon/native-mathml

- Lack of fonts
- No country of mathematicians
- Add-on required
- … otherwise SVG

$$\Gamma(z) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \frac{1}{z+n} + \int_1^{\infty} t^{z-1} e^{-t} dt \,.$$ [4]

$$\Gamma(z) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \frac{1}{z+n} + \int_1^{\infty} t^{z-1} e^{-t} dt \,.$$ [4]

$$\Gamma(z) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \frac{1}{z+n} + \int_1^{\infty} t^{z-1} e^{-t} dt \,.$$ [4]

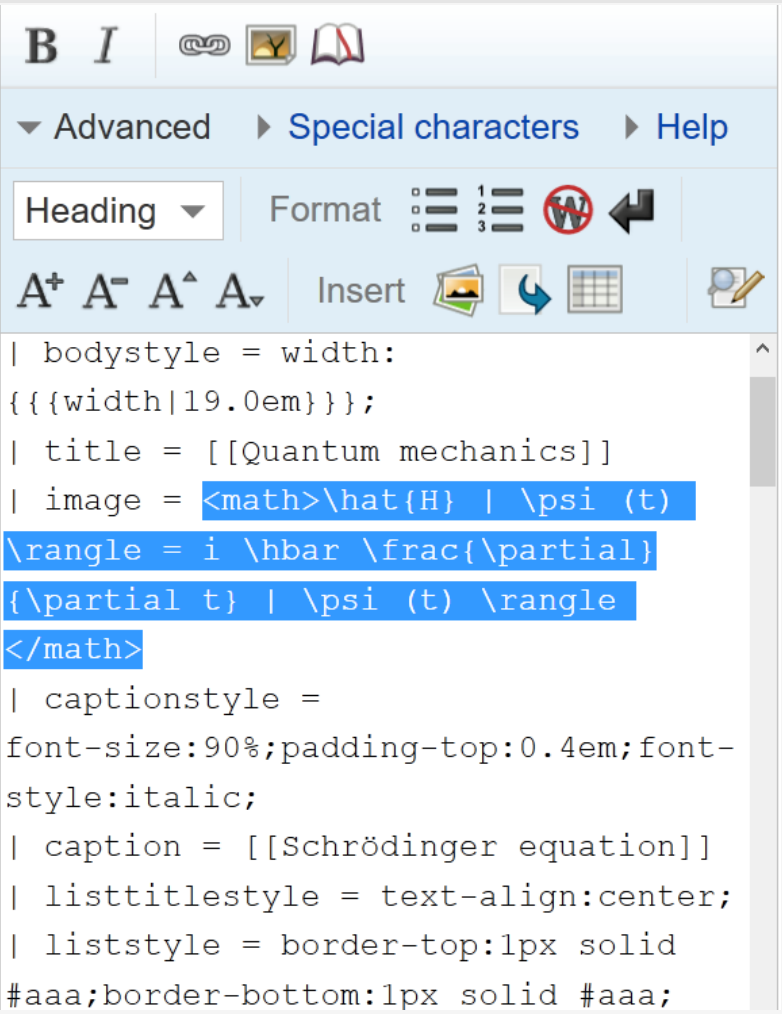
$$\Gamma(z) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \frac{1}{z+n} + \int_1^{\infty} t^{z-1} e^{-t} dt \,.$$ [4]

$$\Gamma(z) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \frac{1}{z+n} + \int_1^{\infty} t^{z-1} e^{-t} dt \,.$$ [4]

Overview

1. Introduction to Wikipedia's new math rendering using mathoid
2. **Improvements over the old math rendering using texvc**
3. Comparing different math rendering engines using mathpipe
    1. Automation Framework
    2. Comparison of math images
4. Future Work

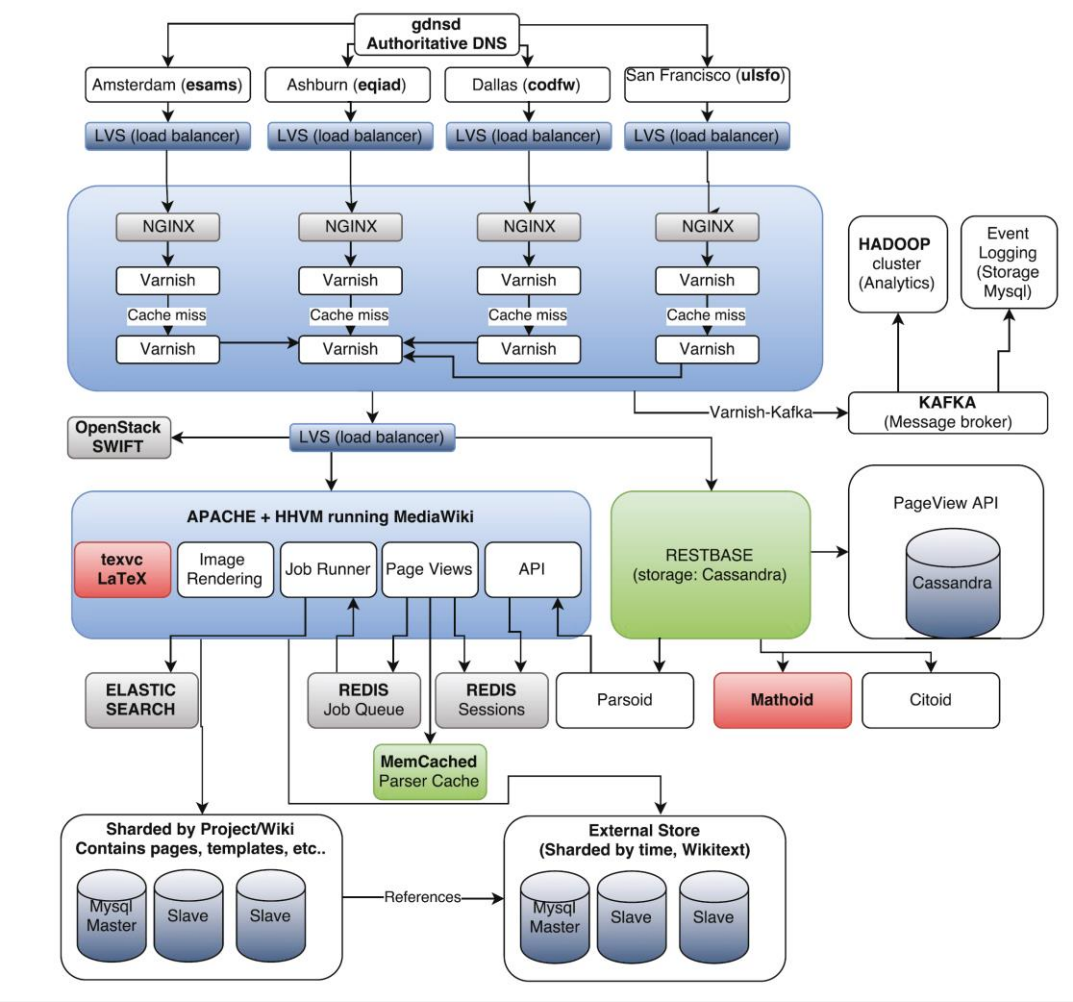
Comparison to old rendering



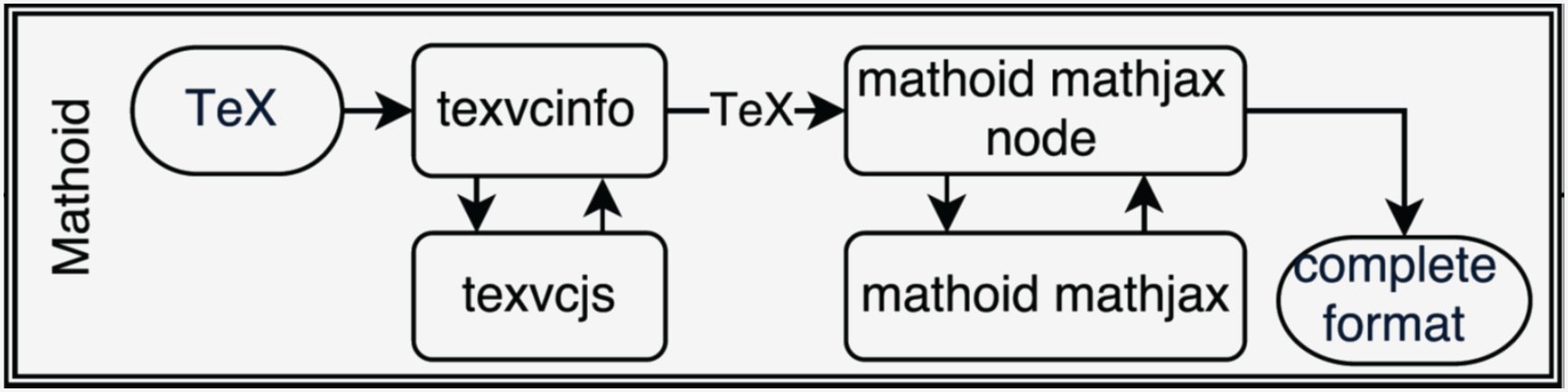
$$i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle$$

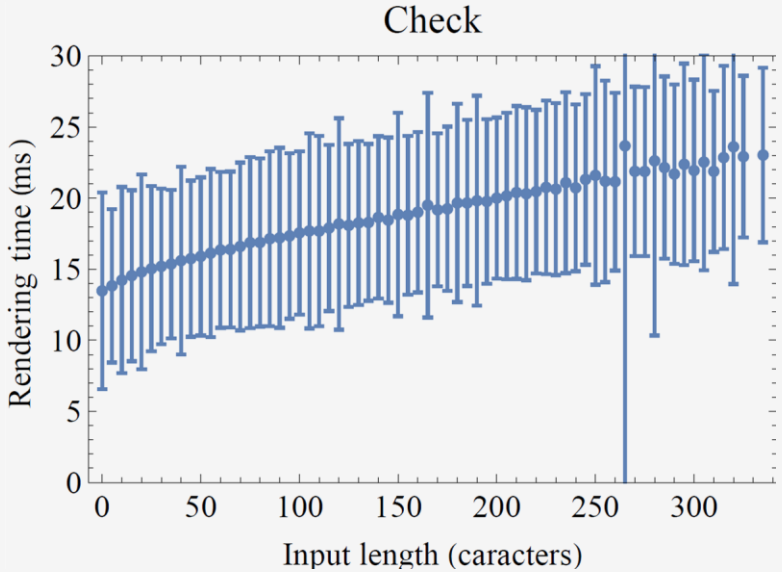
$$i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle$$

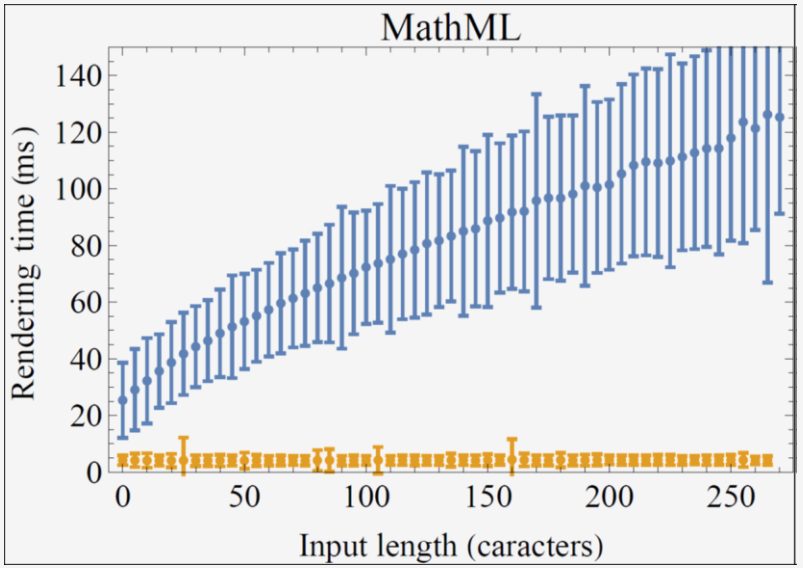
https://github.com/wikimedia/mathoid



Performance



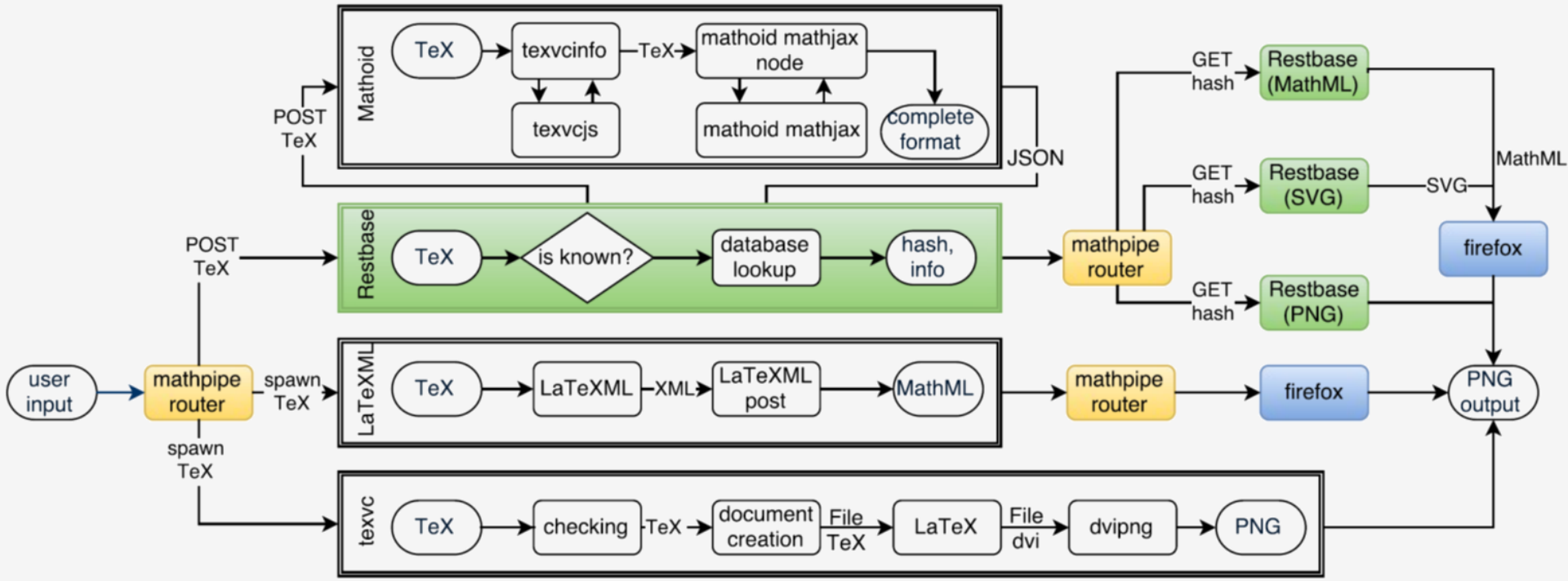




Overview

1. Introduction to Wikipedia's new math rendering using mathoid
2. Improvements over the old math rendering using texvc
3. **Comparing different math rendering engines using mathpipe**
    1. **Automation Framework**
    2. Comparison of math images
4. Future Work

mathpipe Automation



Overview

1. Introduction to Wikipedia's new math rendering using mathoid
2. Improvements over the old math rendering using texvc
3. **Comparing different math rendering engines using mathpipe**
    1. Automation Framework
    2. **Comparison of math images**
4. Future Work

Comparison of math images

$$P = 3B_0 \left(\frac{1-\eta}{\eta^2} \right) e^{\frac{3}{2}(B'_0-1)(1-\eta)}$$

$$P = 3B_0 \left(\frac{1-\eta}{\eta^2} \right) e^{\frac{3}{2}(B'_0-1)(1-\eta)}$$

$$P = 3B_0 \left(\frac{1=\eta}{\eta^2} \right) e^{\frac{3}{2}(B'_0-1)(1-\eta)}$$

Comparison of math images

Significant changes

- Missing characters
- Different shapes
- Space changes
- Spatial misalignments
- Touching and breaking of symbols

Insignificant changes

- Font changes
- Space changes
- Touching and breaking of symbols

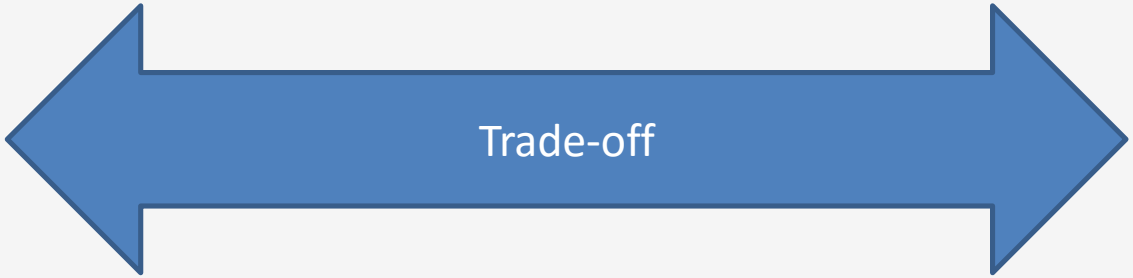
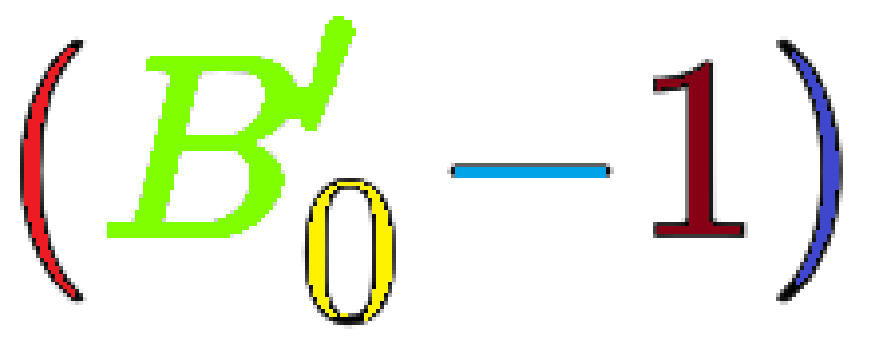


Image comparison method

- Binarisation & Connected component analysis
- Cropping & Scaling
- Simple component pairing
- Touching component analysis
- Result reporting

Binarisation & Connected component analysis

- Binarisation based on threshold
  - Different encodings
  - Grey scale vs. Transparency
- Connected components
  - Standard methods [He et al.]



Cropping & Scaling

- Remove superfluous background padding
- Avoid resize artefacts
- Virtual scaling using floats

$$\lim_{k \to \infty} \| \mathbf{T}^k \| = 0,$$

$$\lim_{k \to \infty} \| \mathbf{T}^k \| = 0,$$

Cropping & Scaling

- Remove superfluous background padding
- Avoid resize artefacts
- Virtual scaling using floats

$$\lim_{k \to \infty} \| \mathbf{T}^k \| = 0,$$

$$\lim_{k \to \infty} \| \mathbf{T}^k \| = 0,$$

Cropping & Scaling

- Remove superfluous background padding
- Avoid resize artefacts
- Virtual scaling using floats

$$\lim_{k \to \infty} \| \mathbf{T}^k \| = 0,$$

$$\lim_{k \to \infty} \| \mathbf{T}^k \| = 0,$$

Cropping & Scaling

- Remove superfluous background padding
- Avoid resize artefacts
- Virtual scaling using floats

$$\lim_{k \to \infty} \| \mathbf{T}^k \| = 0,$$

$$\lim_{k \to \infty} \| \mathbf{T}^k \| = 0,$$

Simple component pairing

Iterative algorithm

1. Tight constraints on
   - position
   - size
   - aspect ratio
2. Verification based on metric shape similarity
3. Relaxing parameters

$$P = 3B_0 \left(\frac{1-\eta}{\eta^2}\right) e^{\frac{3}{2}(B'_0-1)(1-\eta)}$$

$$P = 3B_0 \left(\frac{1-\eta}{\eta^2}\right) e^{\frac{3}{2}(B'_0-1)(1-\eta)}$$

Possible results

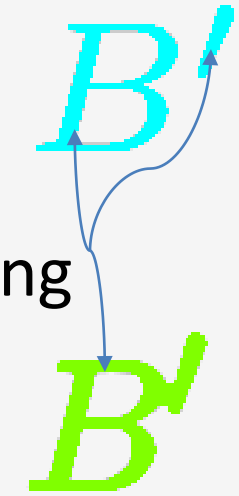
a) No unpaired components

b) Relaxing parameters exceeds limit

c) Multi match

Touching component analysis

- Iterative algorithm
  - Start at largest unpaired component
  1. Try all combinations based on neighbouring unmatched components
  2. Shape similarity test
  3. If success → remove element from set
  - Finish on no remaining elements == finished processing the smallest element



Result reporting

- Components grouped into four groups
  1. Simple matches
  2. Touching matches
  3. First image unpaired
  4. Second image unpaired
- Comparison passed if only groups 1 (+2) are not empty

$$P = 3B_0 \left(\frac{1 = \eta}{\eta^2} \right) e^{\frac{3}{2}((B'_0 - 1)((1 - \eta)}$$

Overview

1. Introduction to Wikipedias new math rendering using mathoid
2. Improvements over the old math rendering using texvc
3. Comparing different math rendering engines using mathpipe
    1. Automation Framework
    2. Comparison of math images

→ 1st **Result**

4. Future Work

1st Result

$$R_{ix}(t) = M_i A_{ix}(t)$$

$$R_{ix}(t) = M_i A_{ix}(t)\,.$$

R_{ix}(t)=M_{i}A_{ix}(t)\frac{}{}

```
<mfrac xml:id="p1.1.m1.1.14a" xref
    <mi xml:id="p1.1.m1.1.14.2" />
    <mi xml:id="p1.1.m1.1.14.3" />
  </mfrac>
```

```
<apply xml:id="p1.1.m1.1.14.cmml" xref="
  <divide xml:id="p1.1.m1.1.14.1.cmml"/>
  <csymbol cd="latexml">absent</csymbol>
  <csymbol cd="latexml">absent</csymbol>
```

Overview

1. Introduction to Wikipedia's new math rendering using mathoid
2. Improvements over the old math rendering using texvc
3. Comparing different math rendering engines using mathpipe
    1. Automation Framework
    2. Comparison of math images
4. **Future Work**

Future work

- Improve reliability
- Investigate all results
- Complete web service

http://png.formulasearchengine.com

- Integrate to the standard WikiMedia development workflow

Contact

Moritz Schubotz (now at Universität Konstanz)

moritz@schubotz.de

+49 7531 88 4438

Mobile: +49 1578 047 1397

www.isg.uni-konstanz.de

www.formulasearchengine.com