



Linus Torvalds

Wykład popularny
dla młodzieży szkół średnich

Wolne oprogramowanie

czyli
czy można żyć bez PowerPointa

Ryszard Tanaś

<http://zon8.physd.amu.edu.pl/~tanas>

2 października 2004

Spis treści

1	Wolne Oprogramowanie	5
1.1	Czym jest?	5
1.2	Linux	7
1.3	Najbardziej znane programy WO (FOSS)	8
2	Czy można żyć bez PowerPointa?	10
3	OpenOffice	10
4	PDF i L^AT_EX	11
4.1	Dlaczego?	11
4.2	Co potrzebujemy?	13
4.3	Jak to zrobić?	14

4.3.1	Czytamy dokumentację	14
4.4	Próbujemy sami	15
4.4.1	Tekst za mgłą	15
4.4.2	Zmieniający się tekst	17
4.4.3	Do przodu do tyłu	18
4.4.4	Wzory matematyczne	19
4.4.5	Grafika — kwazi-animacje	22
4.4.6	Geometria na sferze	28
4.4.7	Uruchamiamy aplikację zewnętrzną	29

1 Wolne Oprogramowanie

1.1 Czym jest?

Wolne oprogramowanie to kwestia wolności, nie ceny. By zrozumieć tę koncepcję, powinniśmy myśleć o wolności słowa, a nie darmowym piwie (angielskie **free** znaczy najczęściej **wolny, swobodny**, ale może też oznaczać **darmowy**).

1 Wolne Oprogramowanie

1.1 Czym jest?

Wolne oprogramowanie to kwestia wolności, nie ceny. By zrozumieć tę koncepcję, powinniśmy myśleć o wolności słowa, a nie darmowym piwie (angielskie **free** znaczy najczęściej **wolny, swobodny**, ale może też oznaczać **darmowy**).

Wolne oprogramowanie odnosi się do prawa użytkowników do swobodnego uruchamiania, kopiowania, rozpowszechniania, analizowania, zmian i ulepszania programów.

1 Wolne Oprogramowanie

1.1 Czym jest?

Wolne oprogramowanie to kwestia wolności, nie ceny. By zrozumieć tę koncepcję, powinniśmy myśleć o wolności słowa, a nie darmowym piwie (angielskie **free** znaczy najczęściej **wolny, swobodny**, ale może też oznaczać **darmowy**).

Wolne oprogramowanie odnosi się do prawa użytkowników do swobodnego uruchamiania, kopiowania, rozpowszechniania, analizowania, zmian i ulepszania programów.

Dokładniej, mówimy o **czterech rodzajach wolności** użytkowników programu:

- wolność uruchamiania programu, w dowolnym celu (wolność 0),
- wolność analizowania, jak program działa, i dostosowywania go do swoich potrzeb (wolność 1). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.
- wolność rozpowszechniania kopii, byście mogli pomóc sąsiadom (wolność 2)
- wolność udoskonalania programu i publicznego rozpowszechniania własnych ulepszeń, dzięki czemu może z nich skorzystać cała społeczność (wolność 3). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.

- wolność uruchamiania programu, w dowolnym celu (wolność 0),
- wolność analizowania, jak program działa, i dostosowywania go do swoich potrzeb (wolność 1). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.
- wolność rozpowszechniania kopii, byście mogli pomóc sąsiadom (wolność 2)
- wolność udoskonalania programu i publicznego rozpowszechniania własnych ulepszeń, dzięki czemu może z nich skorzystać cała społeczność (wolność 3). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.

- wolność uruchamiania programu, w dowolnym celu (wolność 0),
- wolność analizowania, jak program działa, i dostosowywania go do swoich potrzeb (wolność 1). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.
- wolność rozpowszechniania kopii, byście mogli pomóc sąsiadom (wolność 2)
- wolność udoskonalania programu i publicznego rozpowszechniania własnych ulepszeń, dzięki czemu może z nich skorzystać cała społeczność (wolność 3). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.

- wolność uruchamiania programu, w dowolnym celu (wolność 0),
- wolność analizowania, jak program działa, i dostosowywania go do swoich potrzeb (wolność 1). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.
- wolność rozpowszechniania kopii, byście mogli pomóc sąsiadom (wolność 2)
- wolność udoskonalania programu i publicznego rozpowszechniania własnych ulepszeń, dzięki czemu może z nich skorzystać cała społeczność (wolność 3). Warunkiem koniecznym jest tu dostęp do kodu źródłowego.

1.2 Linux

W 1991 r. **Linus Torvalds**, student Uniwersytetu w Helsinkach, korzystając z systemu **Minix** jako wzorca, stworzył jądro systemu operacyjnego **Linux** (nazwa pochodzi od **Linux Is Not Unix**).

1.2 Linux

W 1991 r. **Linus Torvalds**, student Uniwersytetu w Helsinkach, korzystając z systemu **Minix** jako wzorca, stworzył jądro systemu operacyjnego **Linux** (nazwa pochodzi od **Linux Is Not Unix**).

Tego właśnie brakowało w systemie GNU!

1.2 Linux

W 1991 r. **Linus Torvalds**, student Uniwersytetu w Helsinkach, korzystając z systemu **Minix** jako wzorca, stworzył jądro systemu operacyjnego **Linux** (nazwa pochodzi od **Linux Is Not Unix**).

Tego właśnie brakowało w systemie GNU!

Połączenie jądra Linuksa z istniejącymi aplikacjami GNU dało system **GNU/Linux** i zapoczątkowało jego burzliwy rozwój, który stał się symbolem i fenomenem **wolnego oprogramowania**.

1.2 Linux

W 1991 r. **Linus Torvalds**, student Uniwersytetu w Helsinkach, korzystając z systemu **Minix** jako wzorca, stworzył jądro systemu operacyjnego **Linux** (nazwa pochodzi od **Linux Is Not Unix**).

Tego właśnie brakowało w systemie GNU!

Połączenie jądra Linuksa z istniejącymi aplikacjami GNU dało system **GNU/Linux** i zapoczątkowało jego burzliwy rozwój, który stał się symbolem i fenomenem **wolnego oprogramowania**.

Do rozwoju tego przyczyniła się **eksplozja internetu** w latach 1993-1994.

1.3 Najbardziej znane programy WO (FOSS)

- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs, XEmacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUSstep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP, Gnuplot, Octave
- Pakiety biurowe: OpenOffice, KOffice

1.3 Najbardziej znane programy WO (FOSS)

- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs, XEmacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUSstep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP, Gnuplot, Octave
- Pakiety biurowe: OpenOffice, KOffice

1.3 Najbardziej znane programy WO (FOSS)

- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs, XEmacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUSstep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP, Gnuplot, Octave
- Pakiety biurowe: OpenOffice, KOffice

1.3 Najbardziej znane programy WO (FOSS)

- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs, XEmacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUSstep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP, Gnuplot, Octave
- Pakiety biurowe: OpenOffice, KOffice

1.3 Najbardziej znane programy WO (FOSS)

- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs, XEmacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUSstep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP, Gnuplot, Octave
- Pakiety biurowe: OpenOffice, KOffice

1.3 Najbardziej znane programy WO (FOSS)

- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs, XEmacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUSstep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP, Gnuplot, Octave
- Pakiety biurowe: OpenOffice, KOffice

1.3 Najbardziej znane programy WO (FOSS)

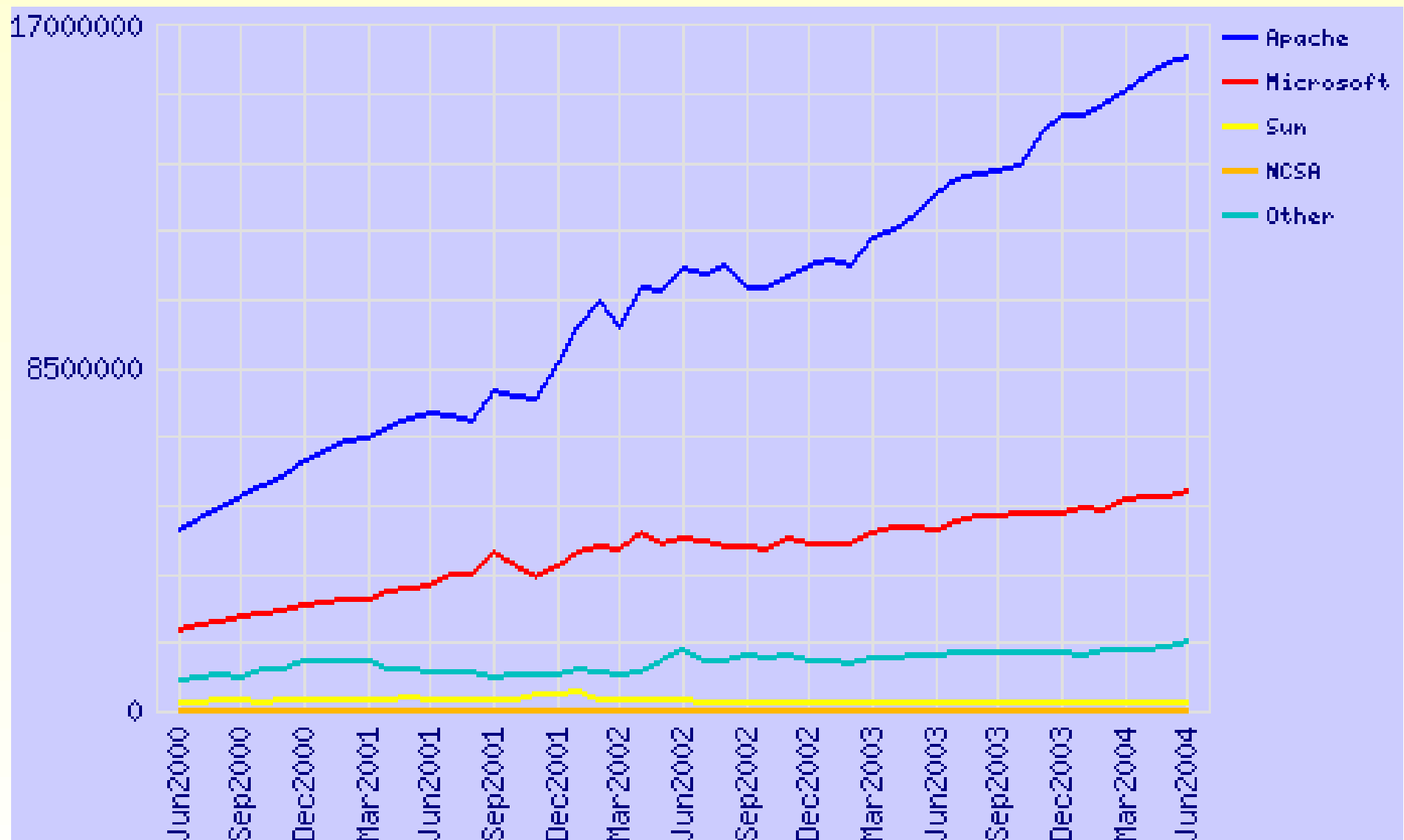
- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs, XEmacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUSstep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP, Gnuplot, Octave
- Pakiety biurowe: OpenOffice, KOffice

1.3 Najbardziej znane programy WO (FOSS)

- Systemy operacyjne: Linux, FreeBSD, OpenBSD, NetBSD, GNU/Hurd
- Języki: GNU C/C++, Perl, Python, TCL
- Edytory: Vi, Emacs, XEmacs
- Systemy okien: The X Window System, XFree86
- Środowiska graficzne: GNOME, KDE, GNUSstep, XFce
- Przeglądarki: Mozilla, Galeon
- Aplikacje: ABIWord, GIMP, Gnuplot, Octave
- Pakiety biurowe: OpenOffice, KOffice

- Oprogramowanie serwerowe: Apache, Samba, PHP, MySQL, PostgreSQL

- Oprogramowanie serwerowe: Apache, Samba, PHP, MySQL, PostgreSQL



Serwery WWW

2 Czy można żyć bez PowerPointa?

Jak przygotować prezentację multimedialną bez PowerPointa?

2 Czy można żyć bez PowerPointa?

Jak przygotować prezentację multimedialną bez PowerPointa?

Pod Linuksem można to zrobić na wiele sposobów!

2 Czy można żyć bez PowerPointa?

Jak przygotować prezentację multimedialną bez PowerPointa?

Pod Linuksem można to zrobić na wiele sposobów!

Patrz np.

<http://www.miwie.org/presentations/presentations.html>

2 Czy można żyć bez PowerPointa?

Jak przygotować prezentację multimedialną bez PowerPointa?

Pod Linuksem można to zrobić na wiele sposobów!

Patrz np.

<http://www.miwie.org/presentations/presentations.html>

3 OpenOffice

Ten program (<http://www.openoffice.org/>) to rozbudowany pakiet biurowy, który daje możliwość przygotowania prezentacji, ale o tym nie dzisiaj. Tę możliwość zostawiam do indywidualnego testowania.

4 PDF i \LaTeX

4.1 Dlaczego?

- **PDF** (Portable Document Format) jest rzeczywiście **przenośny** i można go pokazać na **różnych** platformach
- \LaTeX to **znakomity skład**, zwłaszcza formuł matematycznych
- \LaTeX to pliki tekstowe; możemy używać ulubionego **edytora tekstu**, łatwo poprawić, łatwo przesłać, łatwo przechować
- **PDF** wygląda wszędzie **tak samo**, nie ma problemu kompatybilności

4 PDF i \LaTeX

4.1 Dlaczego?

- PDF (Portable Document Format) jest rzeczywiście przenośny i można go pokazać na różnych platformach
- \LaTeX to znakomity skład, zwłaszcza formuł matematycznych
- \LaTeX to pliki tekstowe; możemy używać ulubionego edytora tekstu, łatwo poprawić, łatwo przesłać, łatwo przechować
- PDF wygląda wszędzie tak samo, nie ma problemu kompatybilności

4 PDF i \LaTeX

4.1 Dlaczego?

- PDF (Portable Document Format) jest rzeczywiście **przenośny** i można go pokazać na **różnych** platformach
- \LaTeX to **znakomity skład**, zwłaszcza formuł matematycznych
- \LaTeX to pliki tekstowe; możemy używać ulubionego **edytora tekstu**, łatwo poprawić, łatwo przesłać, łatwo przechować
- PDF wygląda wszędzie **tak samo**, nie ma problemu kompatybilności

4 PDF i \LaTeX

4.1 Dlaczego?

- PDF (Portable Document Format) jest rzeczywiście **przenośny** i można go pokazać na **różnych** platformach
- \LaTeX to **znakomity skład**, zwłaszcza formuł matematycznych
- \LaTeX to pliki tekstowe; możemy używać ulubionego **edytora tekstu**, łatwo poprawić, łatwo przesłać, łatwo przechować
- PDF wygląda wszędzie **tak samo**, nie ma problemu kompatybilności

- łatwo włączyć grafikę zarówno rastrową jak i wektorową
- PDF jest znacznie mniejszy niż typowa prezentacja PowerPointa; łatwo przesłać, łatwo przechować, można umieścić w witrynie WWW
- PDF łatwo wydrukować

- łatwo włączyć grafikę zarówno rastrową jak i wektorową
- PDF jest znacznie mniejszy niż typowa prezentacja PowerPointa; łatwo przesłać, łatwo przechować, można umieścić w witrynie WWW
- PDF łatwo wydrukować

- łatwo włączyć grafikę zarówno rastrową jak i wektorową
- PDF jest znacznie mniejszy niż typowa prezentacja PowerPointa; łatwo przesłać, łatwo przechować, można umieścić w witrynie WWW
- PDF łatwo wydrukować

4.2 Co potrzebujemy?

- Linux
- \LaTeX , \PDFLaTeX (w każdej dystrybucji Linuksa)
- \TeX Power (<http://texpower.sourceforge.net/>)
Ale są też inne pakiety (patrz np.
<http://www.miwie.org/presentations/>)
- Edytor, np. Emacs + \AUCTeX + \RefTeX
- Xfig (<http://www.xfig.org/>)
- ePiX (<http://mathcs.holycross.edu/~ahwang/current/ePiX.html>)

4.2 Co potrzebujemy?

- Linux
- \LaTeX , \PDFLaTeX (w każdej dystrybucji Linuksa)
- \TeX Power (<http://texpower.sourceforge.net/>)
Ale są też inne pakiety (patrz np.
<http://www.miwie.org/presentations/>)
- Edytor, np. Emacs + \AUCTeX + \RefTeX
- Xfig (<http://www.xfig.org/>)
- ePiX (<http://mathcs.holycross.edu/~ahwang/current/ePiX.html>)

4.2 Co potrzebujemy?

- Linux
- \LaTeX , \PDFLaTeX (w każdej dystrybucji Linuksa)
- \TeXPower (<http://texpower.sourceforge.net/>)
Ale są też inne pakiety (patrz np.
<http://www.miwie.org/presentations/>)
- Edytor, np. Emacs + \AUCTeX + \RefTeX
- Xfig (<http://www.xfig.org/>)
- ePiX (<http://mathcs.holycross.edu/~ahwang/current/ePiX.html>)

4.2 Co potrzebujemy?

- Linux
- \LaTeX , \PDFLaTeX (w każdej dystrybucji Linuksa)
- \TeXPower (<http://texpower.sourceforge.net/>)
Ale są też inne pakiety (patrz np.
<http://www.miwie.org/presentations/>)
- Edytor, np. Emacs + \AUCTeX + \RefTeX
- \Xfig (<http://www.xfig.org/>)
- \ePiX (<http://mathcs.holycross.edu/~ahwang/current/ePiX.html>)

4.2 Co potrzebujemy?

- Linux
- \LaTeX , \PDFLaTeX (w każdej dystrybucji Linuksa)
- \TeX Power (<http://texpower.sourceforge.net/>)
Ale są też inne pakiety (patrz np.
<http://www.miwie.org/presentations/>)
- Edytor, np. Emacs + \AUCTeX + \RefTeX
- Xfig (<http://www.xfig.org/>)
- ePiX (<http://mathcs.holycross.edu/~ahwang/current/ePiX.html>)

4.2 Co potrzebujemy?

- Linux
- \LaTeX , \PDFLaTeX (w każdej dystrybucji Linuksa)
- \TeX Power (<http://texpower.sourceforge.net/>)
Ale są też inne pakiety (patrz np.
<http://www.miwie.org/presentations/>)
- Edytor, np. Emacs + \AUCTeX + \RefTeX
- Xfig (<http://www.xfig.org/>)
- ePiX (<http://mathcs.holycross.edu/~ahwang/current/ePiX.html>)

4.3 Jak to zrobić?

4.3.1 Czytamy dokumentację

<http://www.latex-project.org/>

<http://www.tex.ac.uk/tex-archive/info/beginlatex/beginlatex.a4.pdf>

<http://www.tex.ac.uk/tex-archive/info/latex4wp/latex4wp.pdf>

<http://texpower.sourceforge.net/doc/fulldemo.pdf>

<http://www.sapijaszko.net/pedeefy.pdf>

4.4 Próbujemy sami

4.4.1 Tekst za mgłą

Tekst źródłowy:

```
\liststepwise*{%  
\begin{stepitemize}  
\item Jeden  
\item Dwa  
\item Trzy  
\end{stepitemize}  
}
```

Wynik:

- Jeden
- Dwa
- Trzy

Wynik:

- Jeden
- Dwa
- Trzy

Wynik:

- Jeden
- Dwa
- Trzy

4.4.2 Zmieniający się tekst

Tekst źródłowy:

```
\stepwise*{%  
\steponce{Jeden}  
\steponce{Dwa}  
\steponce{Trzy}  
}
```

Wynik:

Jeden

4.4.2 Zmieniający się tekst

Tekst źródłowy:

```
\stepwise*{%  
\steponce{Jeden}  
\steponce{Dwa}  
\steponce{Trzy}  
}
```

Wynik:

Dwa

4.4.2 Zmieniający się tekst

Tekst źródłowy:

```
\stepwise*{%  
\steponce{Jeden}  
\steponce{Dwa}  
\steponce{Trzy}  
}
```

Wynik:

Trzy

4.4.3 Do przodu, do tyłu

4.4.3 Do przodu, do tyłu

0

4.4.3 Do przodu, do tyłu

0

1

4.4.3 Do przodu, do tyłu

0

1

1

4.4.3 Do przodu, do tyłu

0 1 1 0

4.4.3 Do przodu, do tyłu

0 1 1 0 1

4.4.3 Do przodu, do tyłu

0 1 1 0 1 1

4.4.3 Do przodu, do tyłu

0 1 1 0 1 1 0

4.4.3 Do przodu, do tyłu

0 1 1 0 1 1 0 0

4.4.3 Do przodu, do tyłu

0

1

1

0

1

1

0

0

$0 \cdot 2^0$

4.4.3 Do przodu, do tyłu

0

1

1

0

1

1

0

0

$0 \cdot 2^1$ $0 \cdot 2^0$

4.4.3 Do przodu, do tyłu

0 1 1 0 1 1 0 0

$1 \cdot 2^2$ $0 \cdot 2^1$ $0 \cdot 2^0$

4.4.3 Do przodu, do tyłu

0 1 1 0 1 1 0 0

$1 \cdot 2^3$ $1 \cdot 2^2$ $0 \cdot 2^1$ $0 \cdot 2^0$

4.4.3 Do przodu, do tyłu

0 1 1 0 1 1 0 0

$0 \cdot 2^4$ $1 \cdot 2^3$ $1 \cdot 2^2$ $0 \cdot 2^1$ $0 \cdot 2^0$

4.4.3 Do przodu, do tyłu

0 1 1 0 1 1 0 0

$1 \cdot 2^5$ $0 \cdot 2^4$ $1 \cdot 2^3$ $1 \cdot 2^2$ $0 \cdot 2^1$ $0 \cdot 2^0$

4.4.3 Do przodu, do tyłu

0 1 1 0 1 1 0 0

$1 \cdot 2^6$ $1 \cdot 2^5$ $0 \cdot 2^4$ $1 \cdot 2^3$ $1 \cdot 2^2$ $0 \cdot 2^1$ $0 \cdot 2^0$

4.4.3 Do przodu, do tyłu

0 1 1 0 1 1 0 0

$0 \cdot 2^7$ $1 \cdot 2^6$ $1 \cdot 2^5$ $0 \cdot 2^4$ $1 \cdot 2^3$ $1 \cdot 2^2$ $0 \cdot 2^1$ $0 \cdot 2^0$

4.4.4 Wzory matematyczne

Jak by to zrobić w PowerPoint?

$$\sum_{i=1}^{\left[\frac{n}{2}\right]} \binom{x_{i,i+1}^{i^2}}{\left[\frac{i+3}{3}\right]} \frac{\sqrt{\mu(i)^{\frac{3}{2}}(i^2-1)}}{\sqrt[3]{\rho(i)-2} + \sqrt[3]{\rho(i)-1}} \quad (1)$$

4.4.4 Wzory matematyczne

Jak by to zrobić w PowerPoint?

$$\sum_{i=1}^{\left[\frac{n}{2}\right]} \binom{x_{i,i+1}^{i^2}}{\left[\frac{i+3}{3}\right]} \frac{\sqrt{\mu(i)^{\frac{3}{2}}(i^2-1)}}{\sqrt[3]{\rho(i)-2} + \sqrt[3]{\rho(i)-1}} \quad (1)$$

W \LaTeX u wygląda to tak:

```
\begin{equation}
\sum_{i=1}^{\left[\frac{n}{2}\right]}
\binom{x_{i,i+1}^{i^2}}{\left[\frac{i+3}{3}\right]}
\frac{\sqrt{\mu(i)^{\frac{3}{2}}(i^2-1)}}
{\sqrt[3]{\rho(i)-2} + \sqrt[3]{\rho(i)-1}}
\end{equation}
```

Albo to?

$$\frac{1}{\sqrt{2} + \frac{1}{\sqrt{3} + \frac{1}{\sqrt{4} + \dots}}}$$

(2)

Albo to?

$$\frac{1}{\sqrt{2} + \frac{1}{\sqrt{3} + \frac{1}{\sqrt{4} + \frac{1}{\dots}}}} \quad (2)$$

W \LaTeX u robi się to tak:

```
\begin{equation}
\cfrac{1}{\sqrt{2}+\cfrac{1}{\sqrt{3}+\cfrac{1}{\sqrt{4}+\cfrac{1}{\cdots}}}}
\end{equation}
```

Wzory możemy budować stopniowo

$$C_m^{(2)} =$$

Wzory możemy budować stopniowo

$$C_m^{(2)} = C_m^{(0)}$$

Wzory możemy budować stopniowo

$$C_m^{(2)} = C_m^{(0)} - \frac{i}{\hbar} \sum_n \int_0^t \tilde{V}_{mn}(\tau) d\tau C_n^{(0)}$$

Wzory możemy budować stopniowo

$$\begin{aligned} C_m^{(2)} &= C_m^{(0)} \\ &\quad - \frac{i}{\hbar} \sum_n \int_0^t \tilde{\mathcal{V}}_{mn}(\tau) d\tau C_n^{(0)} \\ &\quad + \left(-\frac{i}{\hbar}\right)^2 \sum_{n,k} \int_0^t \tilde{\mathcal{V}}_{mn}(\tau_1) \int_0^{\tau_1} \tilde{\mathcal{V}}_{nk}(\tau_2) d\tau_2 d\tau_1 C_k^{(0)} \end{aligned}$$

zaznaczając istotne elementy.

Wzory możemy budować stopniowo

$$\begin{aligned} C_m^{(2)} = & C_m^{(0)} \\ & - \frac{i}{\hbar} \sum_n \int_0^t \tilde{\mathcal{V}}_{mn}(\tau) d\tau C_n^{(0)} \\ & + \left(-\frac{i}{\hbar}\right)^2 \sum_{n,k} \int_0^t \tilde{\mathcal{V}}_{mn}(\tau_1) \int_0^{\tau_1} \tilde{\mathcal{V}}_{nk}(\tau_2) d\tau_2 d\tau_1 C_k^{(0)} \end{aligned}$$

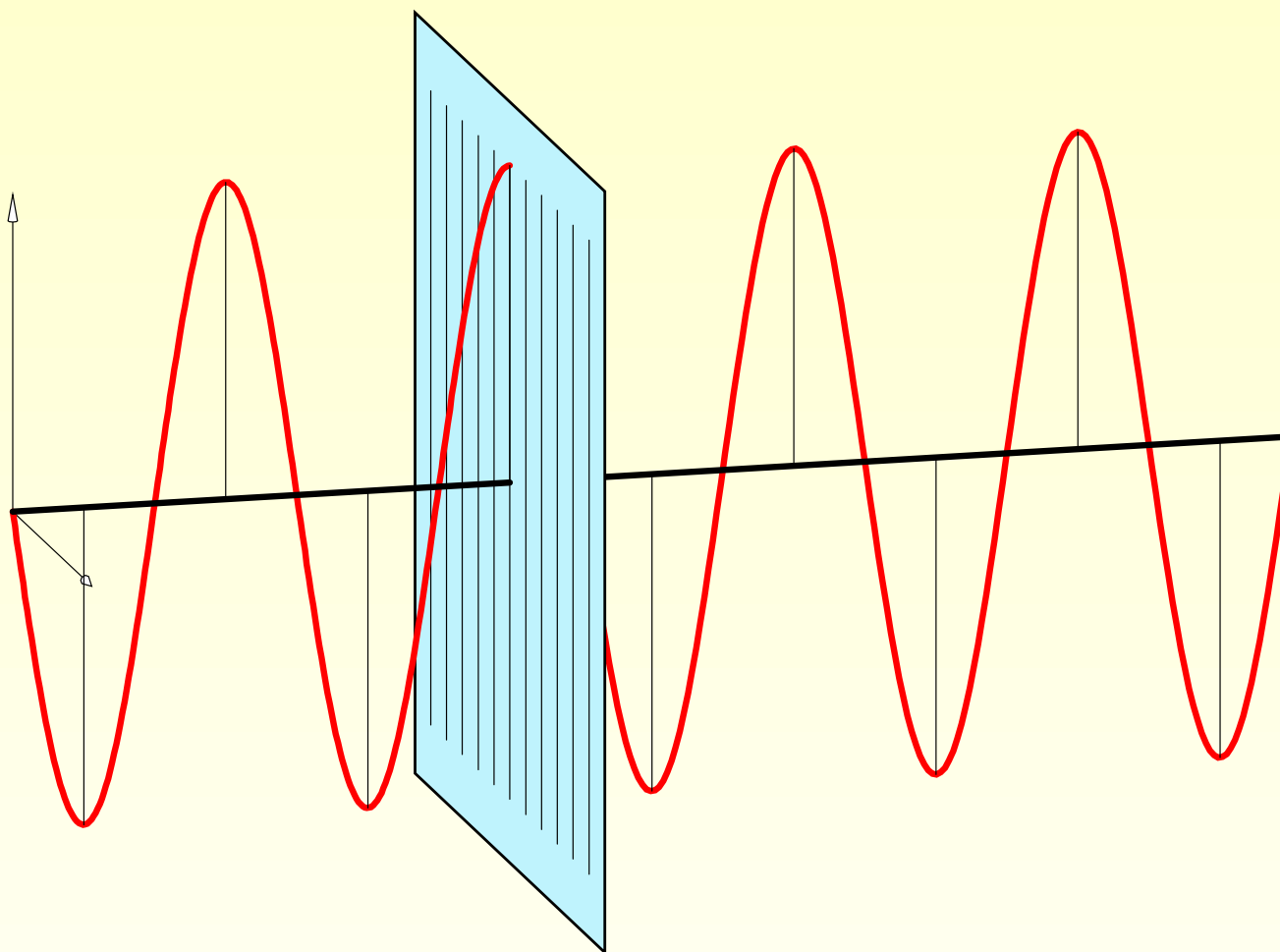
zaznaczając istotne elementy.

Bardziej skomplikowane przykłady patrz:

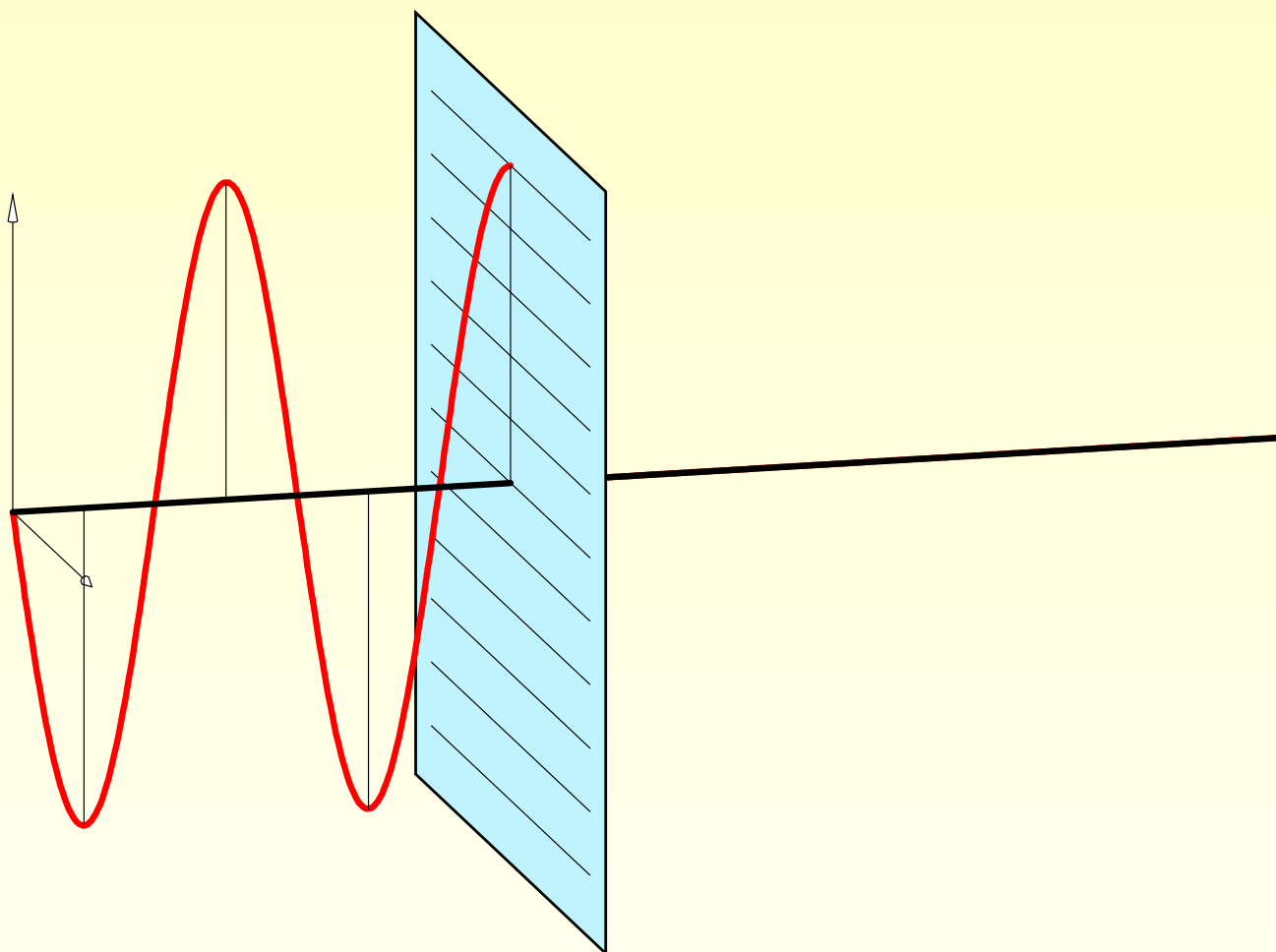
<http://texpower.sourceforge.net/doc/fulldemo.pdf>

4.4.5 Grafika — kwazi-animacje

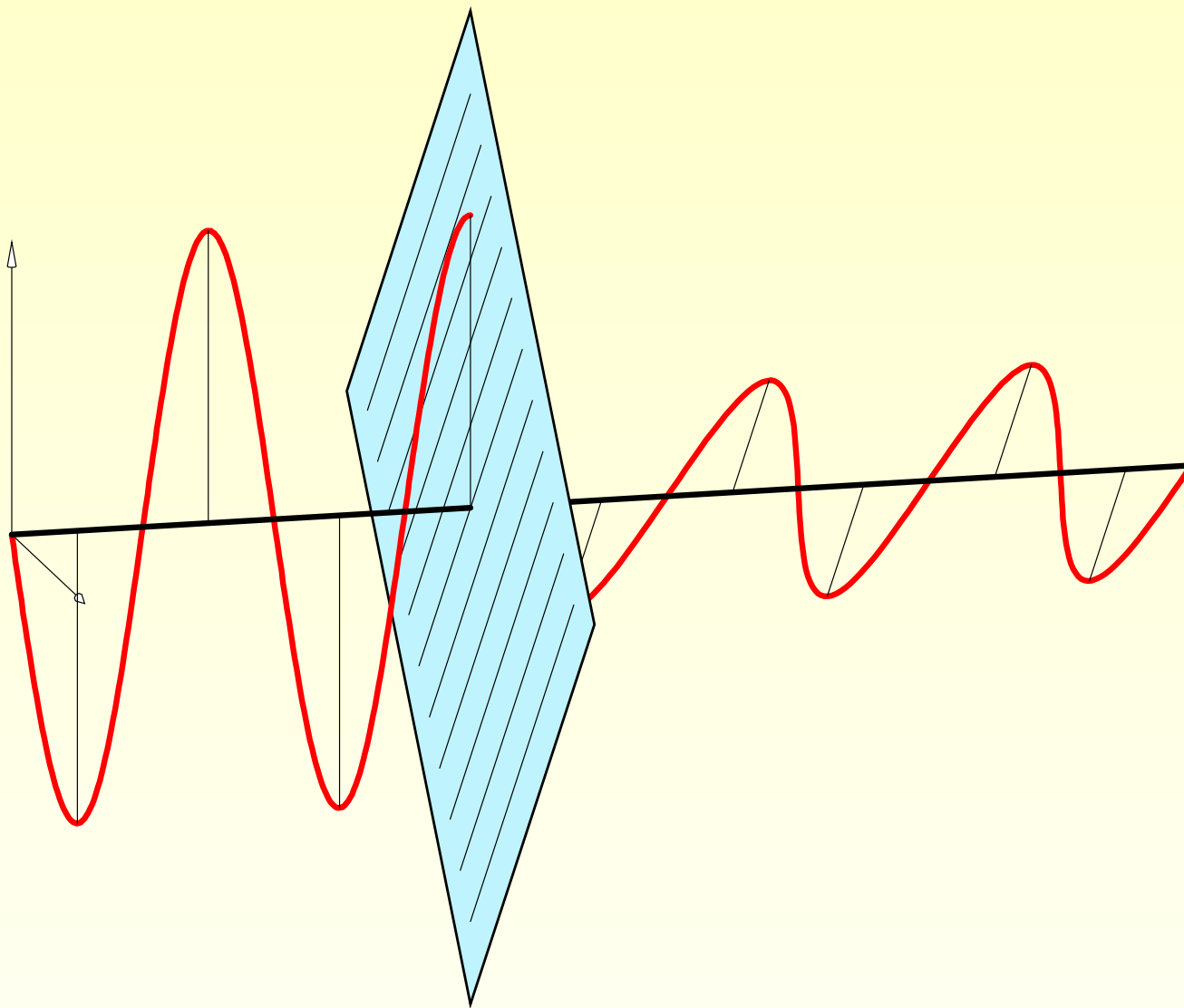
Przygotowujemy sekwencję rysunków jednym z programów graficznych, np. **Xfig** lub **ePiX**. W tym przypadku był to **ePiX**, który pozwala na poprawne matematycznie rzutowanie obiektów trójwymiarowych na płaszczyznę ekranu.



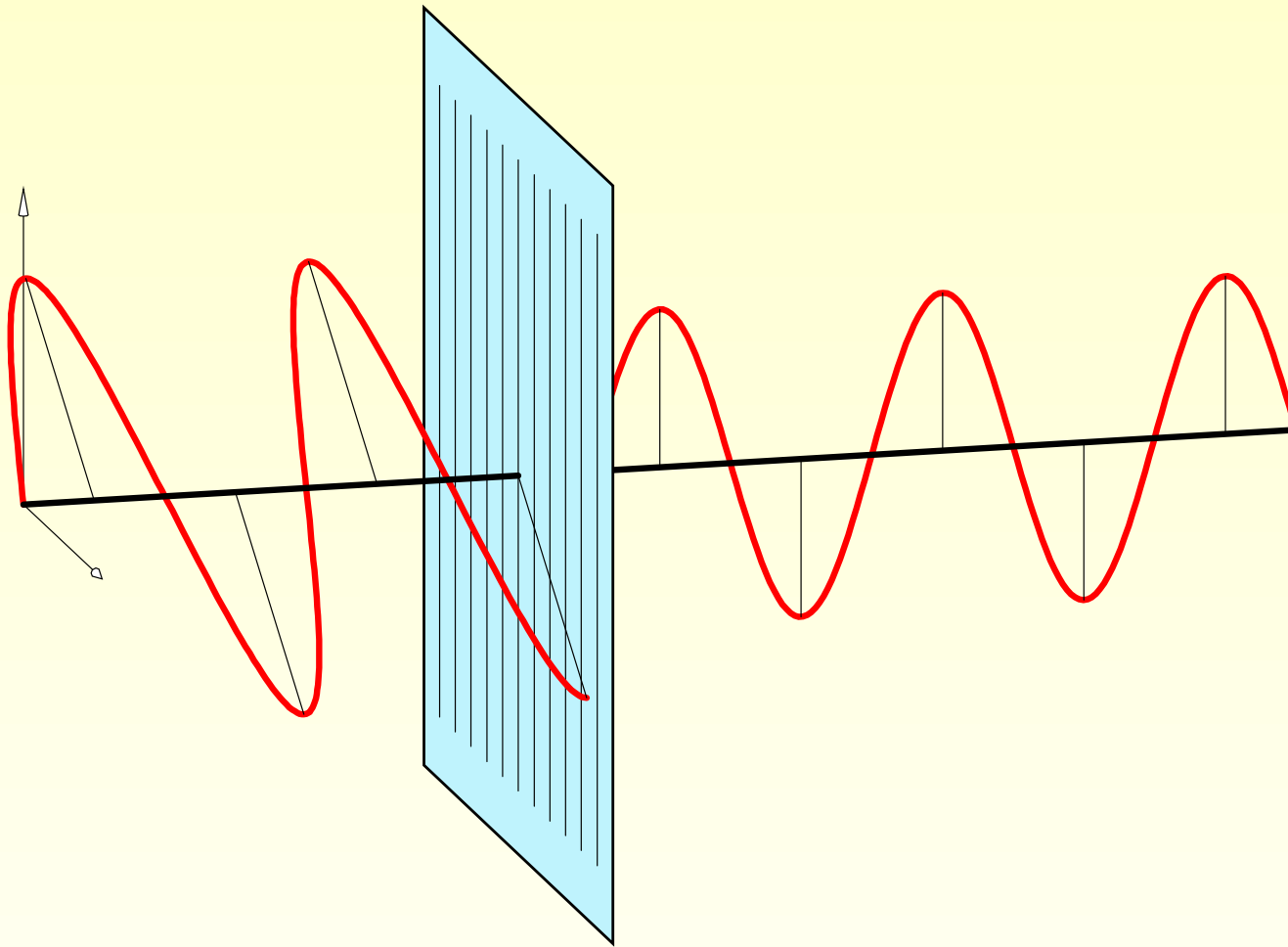
Polaryzator ustawiony pionowo przepuszcza światło spolaryzowane pionowo.



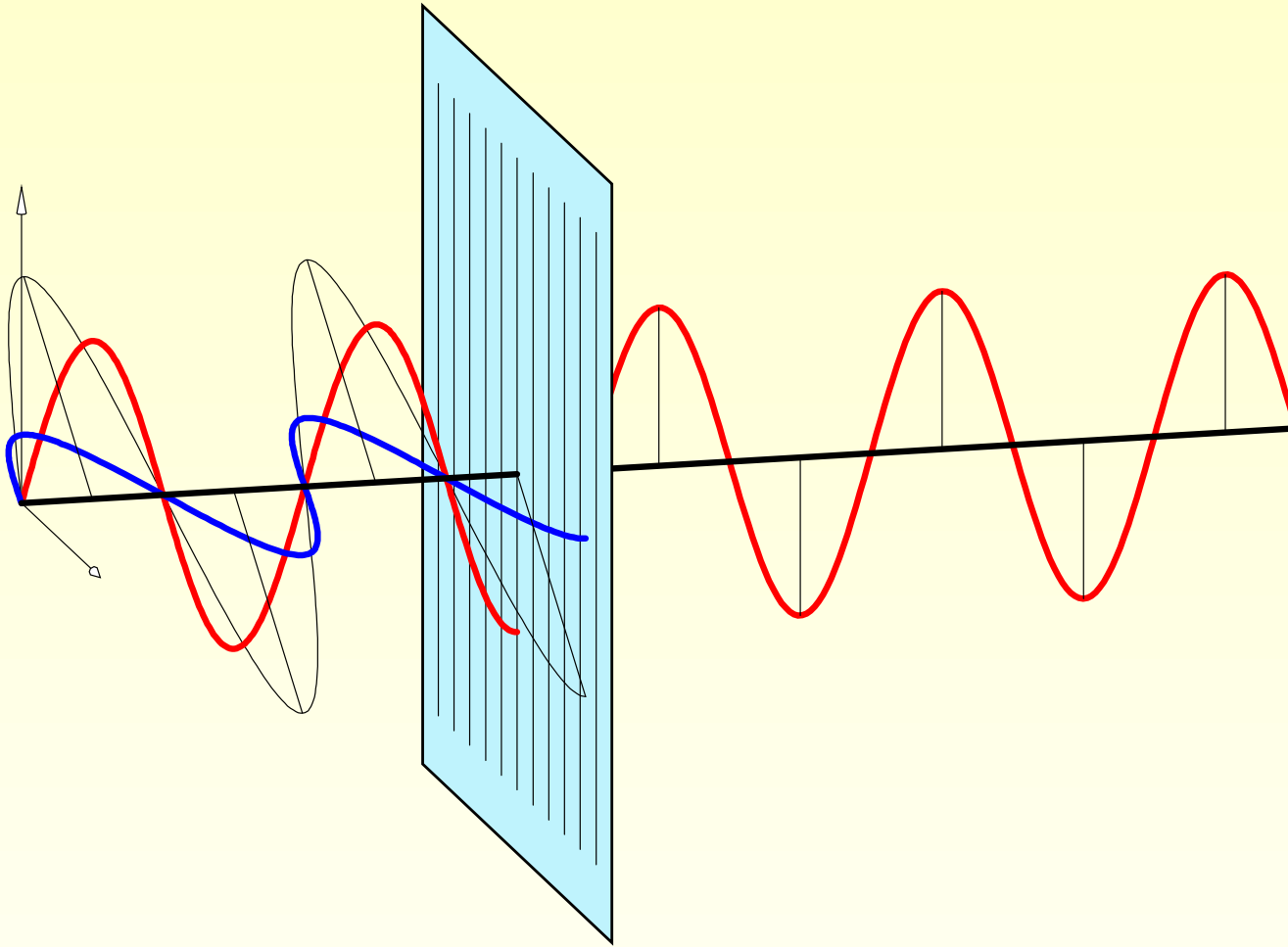
Polaryzator ustawiony **poziomo** zatrzymuje światło spolaryzowane **pionowo**.



Polaryzator ustawiony **ukośnie** przepuszcza światło spolaryzowane **ukośnie**. Skąd się wzięło światło spolaryzowane **ukośnie**?

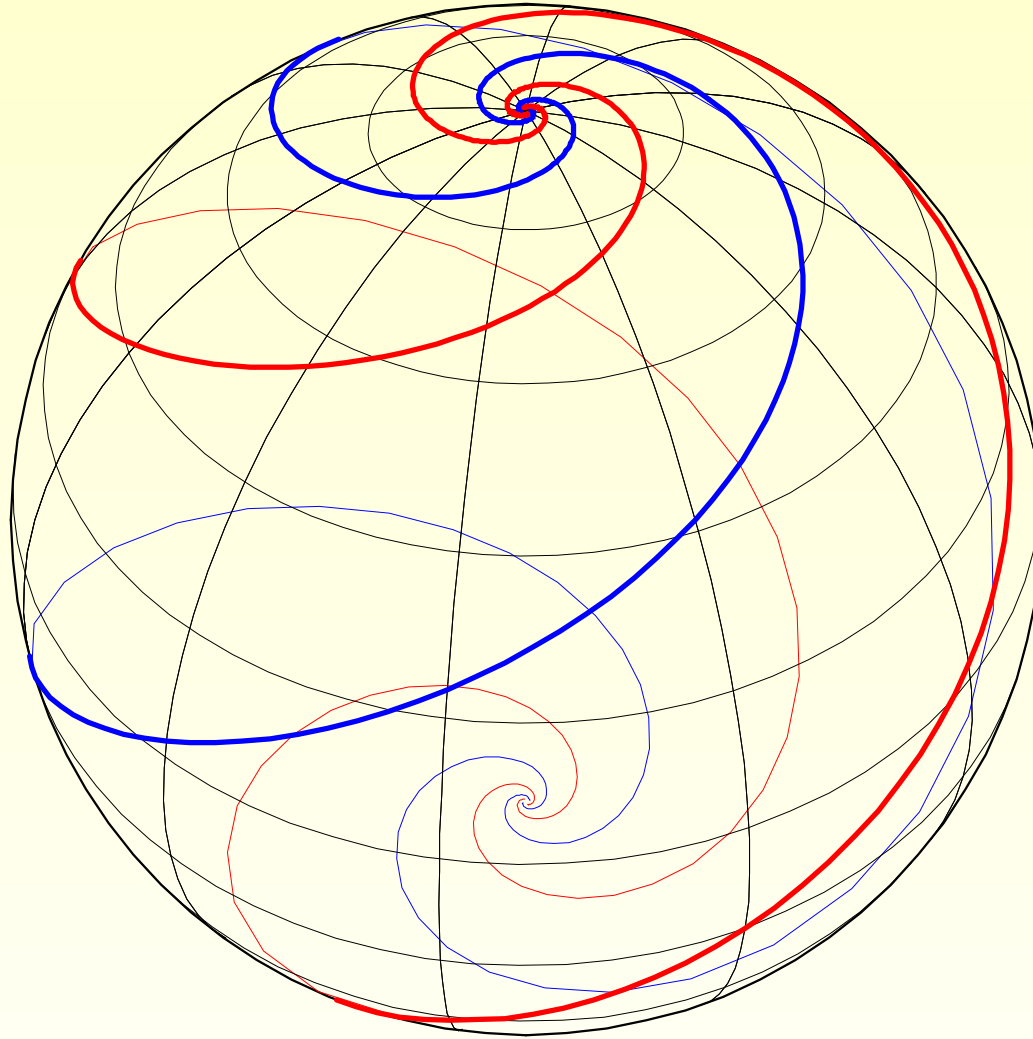


Pada światło spolaryzowane **ukośnie**, polaryzator ustawiony **pionowo** przepuszcza światło spolaryzowane **pionowo**.



Polaryzacja ukośna jest superpozycją polaryzacji pionowej i poziomej. Polaryzator przepuszcza tylko składową pionową!

4.4.6 Geomeria na sferze



Rysunek 1: Przykład `sphere.xp` z ePiXa. Takich rysunków nie da się zrobić „ręcznie”!

4.4.7 Uruchamiamy aplikację zewnętrzną

RSA demo

4.4.7 Uruchamiamy aplikację zewnętrzną

RSA demo

Xfig

4.4.7 Uruchamiamy aplikację zewnętrzną

RSA demo

Xfig

No to wszystko możemy zrobić!

Pliki źródłowe tej prezentacji można znaleźć pod adresem:

<http://zon8.physd.amu.edu.pl/~tanias/lecture.html>

Pliki źródłowe tej prezentacji można znaleźć pod adresem:

<http://zon8.physd.amu.edu.pl/~tanias/lecture.html>

Powodzenia!