Computational Tools for String Phenomenology
OR: The importance of cohomology group dimensions

Benjamin Jurke

Max-Planck-Institut für Physik
(Werner-Heisenberg-Institut)

Contents
- String pheno & cohomology
- Toric geometry 1-0-1
- Algorithm & Applications
- cohomCalc implementation

Boston (Northeastern University) — June 13, 2011

with R. Blumenhagen, T. Rahn, H. Roschy

- Algorithm: arXiv:1003.5217
- Applications: arXiv:1010.3717
Part 1

Motivation:
String phenomenology & cohomology groups
I. Motivation: Type II string models

In Type II string models intersecting D-branes and gauge fluxes are required in order to generate chiral matter.

Example: $U(1)$ gauge flux on D7s

$D_a, D_b$ two stacks of D7s intersecting over curve $C = D_a \cap D_b$, matter in the bifundamental $(\bar{N}_a, N_b)$ is counted by

$$H^i(C; L^\vee_a \otimes L_b \otimes K^{\frac{1}{2}}_C).$$

The chiral index gives the net number of chiral states:

$$I_{ab}^{loc} = \chi(C; L^\vee_a \otimes L_b \otimes K^{\frac{1}{2}}_C) = \int_X [D_a] \wedge [D_b] \wedge (c_1(L_a) - c_1(L_b))$$

[Blumenhagen-Körs-Lüst-Stieberger '06, Blumenhagen-Braun-Grimm-Weigand '08]
Euclidean D-brane instantons (E-branes) are entirely wrapped around the compactified dimensions, i.e. pointlike from the 4d perspective.

**Example: Zero modes counting for E3-brane instanton**

<table>
<thead>
<tr>
<th>zero modes</th>
<th>number</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_\mu$, $\theta_\alpha$</td>
<td>1</td>
</tr>
<tr>
<td>$\bar{\tau}_{\bar{\alpha}}$</td>
<td>0</td>
</tr>
<tr>
<td>$\gamma_\alpha$</td>
<td>$h^{1,0}_+ (E)$</td>
</tr>
<tr>
<td>$w$, $\bar{\gamma}_{\bar{\alpha}}$</td>
<td>$h^{1,0}_- (E)$</td>
</tr>
<tr>
<td>$\chi_\alpha$</td>
<td>$h^{2,0}_+ (E)$</td>
</tr>
<tr>
<td>$c$, $\bar{\chi}_{\bar{\alpha}}$</td>
<td>$h^{2,0}_- (E)$</td>
</tr>
</tbody>
</table>

Self-invariant $O(1)$ instanton

[Blumenhagen-Cvetic-Kachru-Weigand '08]
I. Motivation: E3/M5-instanton matching

Type IIB E3-brane instanton zero modes can be matched to vertical M5-brane instantons in F-theory.

Example: Zero modes matching for E3/M5-brane instantons
I. Motivation: Heterotic string models

In heterotic string modes on the Calabi-Yau 3-fold $\mathcal{X}$ a (stable, holomorphic) vector bundle breaks the gauge group $E_8 \times E_8$ or $SO(32)$.

Example: $SU(5)$ model in heterotic string theory

Let $\mathcal{X} = \mathbb{P}^4[5]$ be the quintic Calabi-Yau 3-fold. The monad

$$
0 \longrightarrow V \longrightarrow \mathcal{O}_\mathcal{X}(2)^{\oplus 5} \oplus \mathcal{O}_\mathcal{X}(1)^{\oplus 5} \longrightarrow \mathcal{O}_\mathcal{X}(3)^{\oplus 5} \longrightarrow 0
$$

describes an $SU(5)$-bundle, yielding a $SU(5)$ GUT of the $E_8 \times E_8$ heterotic string (after modding out $\mathbb{Z}_5 \times \mathbb{Z}_5$). The particle spectrum is given by:

- $\eta_{10} = h^1(\mathcal{X}; V)$
- $\eta_{10} = h^1(\mathcal{X}; V^*)$
- $\eta_5 = h^1(\mathcal{X}; \Lambda^2 V)$
- $\eta_5 = h^1(\mathcal{X}; \Lambda^2 V^*)$

[Anderson-Gray-He-Lukas '09]
I. Motivation: Why line bundle cohomology?

In all mentioned examples the phenomenological aspects are ultimately determined by the dimension of vector-bundle-valued cohomology groups.

In fact: Most vector bundles are constructed as monads from line bundles

\[ 0 \rightarrow V \leftarrow \bigoplus_j \mathcal{O}_X(b_j) \rightarrow \bigoplus_k \mathcal{O}_X(c_k) \rightarrow 0 \]

Examples of monads

- The **tangent bundle** \( T_X \) of toric varieties is a monad.
- The **vector bundle moduli** can be computed from \( \text{End}(V) \cong V \otimes V^* \).

Important tool: **Exactness of sequences**, i.e. \( \text{image}(f_i) = \text{kernel}(f_{i+1}) \).

If a sequence is **exact**, the location of dimension-0 spaces often suffices to determine isomorphisms, which makes computations a lot easier.
I. Motivation: Why line bundle cohomology?

From short exact sequences of sums of line bundles one considers the induced long exact sequences of the cohomology, e.g.:

\[ 0 \to \mathcal{O}_X^{\oplus r} \to \bigoplus_k \mathcal{O}_X(D_k) \to T_X \to 0 \]

\[ \downarrow \]

\[ 0 \to \bigoplus_k H^0(X; \mathcal{O}_X(D_k)) \to H^0(X; T_X) \to \bigoplus_k H^1(X; \mathcal{O}_X(D_k)) \to H^1(X; T_X) \to \bigoplus_k H^2(X; \mathcal{O}_X(D_k)) \to H^2(X; T_X) \to \ldots \]
I. Motivation: ...resumé?!

From a phenomenologists point of view, everything boils down to the computation of line bundle-valued cohomology group dimension $h^i(X; L_X)$. What to do?

**Known methods**

- **Isomorphisms**: If you can find isomorphisms to spaces with known cohomology, you don’t have to compute anything.

- **Spectral sequences**: The method of spectral sequences allows to compute the cohomology of general spaces, but it is extremely laborious to work with.

→ Find a new method to compute line bundle cohomology!
Part 2

- Basics of toric geometry

My kind of "Toric Variety."
Reminder: Complex projective space

\( \mathbb{P}^n \) is constructed as a quotient space. The coordinates \( x_0, \ldots, x_n \) of \( \mathbb{C}^{n+1} \) are subject to the \( \mathbb{C}^\times \)-action

\[
(x_0, \ldots, x_n) \mapsto (\lambda x_0, \ldots, \lambda x_n) \quad \text{for all } \lambda \in \mathbb{C}^\times,
\]

if the origin \( 0 \in \mathbb{C}^{n+1} \) is removed, thus \( \mathbb{P}^n = \frac{\mathbb{C}^{n+1} - \{0\}}{\mathbb{C}^\times} \).

Weighted projective space

Generalize the \( \mathbb{C}^\times \)-action by changing the powers of \( \lambda \):

\[
(x_0, \ldots, x_n) \mapsto (\lambda^{Q_0} x_0, \ldots, \lambda^{Q_n} x_n) \quad \text{for all } \lambda \in \mathbb{C}^\times,
\]

giving \( \mathbb{P}^n_{Q_0, \ldots, Q_n} \). Note that \( \mathbb{P}^n = \mathbb{P}^n_{1, \ldots, 1} \).
Use several $\mathbb{C}^\times$-actions with different weights. The powers are given by charges $Q^a_k$ such that

$$
(x_0, \ldots, x_n) \mapsto \left( (\lambda_1^{Q^1_0} \cdots \lambda_r^{Q^r_0}) x_0, \ldots, (\lambda_1^{Q^1_n} \cdots \lambda_r^{Q^r_n}) x_n \right) \quad \text{for } \lambda_i \in \mathbb{C}^\times,
$$

defines a $(\mathbb{C}^\times)^r$-action. Instead of just the origin, a set $Z$ has to be removed from $\mathbb{C}^{n+r}$.

The **toric variety** is then

$$
\mathcal{X} = \frac{\mathbb{C}^{n+r} - Z}{(\mathbb{C}^\times)^r}
$$

There is an alternative, entirely combinatorial perspective on toric varieties that makes this kind of geometry ideally suited for algorithmic treatments.
Stanley-Reisner ideal

The Stanley-Reisner ideal encodes sets of coordinates $x_i$ that are **NOT** allowed to vanish simultaneously. It can be generated by squarefree monomials $S_j$, which are the products of those coordinates:

$$SR(X) = \{x_{k_1} \cdots x_{k_p} : \{x_{k_1}, \ldots, x_{k_p}\} \in \mathbb{Z}\} = \langle S_1, \ldots, S_t \rangle$$

Example: $SR(\mathbb{P}^2) = \langle x_0 x_1 x_2 \rangle$ encondes the removed origin.

Divisors on toric varieties

A divisor on a toric variety is basically a **formal sum of codimension-1 hypersurfaces**. With respect to the chosen coordinates $x_0, \ldots, x_n$ of $X$, one frequently uses the divisors $D_i := \{x_i = 0\} \subset X$. 
II. Toric Geometry 1-0-1: Line bundles

From the projective powers $Q_k^a$ of a toric variety (called GLSM charges in a different context) one can directly read off the classes of the divisors $D_i$.

Example: $\mathbb{P}^2 = \mathbb{P}_{1,1,1}^2$ has just one projective relation and each power is $Q_{1}^1 = 1$. Thus $[D_1] = [D_2] = [D_3] = H$.

Example: The del Pezzo-1 surface (single blowup of $\mathbb{P}^2$) has two projective relations:

<table>
<thead>
<tr>
<th>coords</th>
<th>powers</th>
<th>divisor class</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$Q^1$</td>
<td>$H$</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$Q^2$</td>
<td>$H$</td>
</tr>
<tr>
<td>$x_3$</td>
<td>$Q^1$</td>
<td>$H + X$</td>
</tr>
<tr>
<td>$x_4$</td>
<td>$Q^2$</td>
<td>$X$</td>
</tr>
</tbody>
</table>

Divisors & line bundles

Line bundles $\mathcal{O}(D)$ and divisor $D$ are in a direct correspondence:

$\text{divisor classes} \cong \text{line bundle classes}$
What now?

Part 3

- An **algorithm** to compute line bundle-valued cohomology group dimensions

Remember: Ultimately, we are interested in computing \( \dim H^i(X; O_X(D)) \).

[2 × Blumenhagen-Jurke-Rahn-Roschy '10, Rahn-Roschy '10, Jow '10]
III. Algorithm: Description

Input data: toric variety $X$

- homogeneous coordinates $H = \{x_1, \ldots, x_n\}$
- associated GLSM charges $Q^a_i$ for each $x_i$
- Stanley-Reisner ideal $SR = \langle S_1, \ldots, S_N \rangle$

Basic idea of the algorithm

**Count monomials** of a specific form which carry the same GLSM charge as the divisor $D$ specifying the line bundle $\mathcal{O}_X(D)$.

But: One also has to determine to which cohomology group dimension $h^i(X; \mathcal{O}_X(D))$ (i.e. to which degree $i$) this counting contributes.
III. Algorithm: Description

Take a squarefree monomial $Q = x_{i_1} \cdots x_{i_k}$ of the coordinates $H$.

Consider monomials of the form

$$R^Q(x_1, \ldots, x_n) = \frac{T(x_{j_1}, \ldots, x_{j_{n-k}})}{x_{i_1} \cdots x_{i_k}} \cdot W(x_{i_1}, \ldots, x_{i_k})$$

$T, W$ monomials

$$= x_{j_1}^{\rho_{j_1}} \cdots x_{j_{n-k}}^{\rho_{j_{n-k}}} \cdot x_{i_1}^{-1-\rho_{i_1}} \cdots x_{i_k}^{-1-\rho_{i_k}}$$

- Coordinates in $Q$ have negative powers $\leq -1$
- Remaining coordinates in $H \setminus Q$ have positive powers $\geq 0$
III. Algorithm: Description

Step 1: Count number of monomials with degree of $D$

$$\mathcal{N}_D(Q) := \dim \{ R^Q : \deg_{\text{GLSM}} R^Q = D \}$$

$Q =$ squarefree monomial from the union of the coordinates in SR ideal generators

Determine to which cohomology group dimension $h^i(X; \mathcal{O}_X(D))$ the number $\mathcal{N}_D(Q)$ contributes.

Trace back how often the same $Q$ arises.

For each $Q$ build up an abstract simplex $\Gamma^Q := \{ S \subset SR : Q(S) = Q \}$ with $k$-faces

$$F_k(\Gamma^Q) := \{ S \in \Gamma^Q : |S| = k + 1 \}.$$
III. Algorithm: Description

\[ \phi_k : F_k(\Gamma^Q) \rightarrow F_{k-1}(\Gamma^Q) \]

\[ e_\rho \mapsto \sum_{s \in \rho} \text{sign}(s, \rho) e_{\rho-\{s\}} \]

defines the boundary mappings, where \( e_{\rho-\{s\}} = 0 \) if \( e_{\rho-\{s\}} \notin \Gamma^Q \).

Step 2: Multiplicity factor and group contribution

Consider the (reduced) homology \( \tilde{H}_*(\Gamma^Q) \) and define the multiplicity factors

\[ h_i(Q) := \dim \tilde{H}_{|Q|-i-1}(\Gamma^Q) \]

Those multiplicity factors are 0 or 1 in most cases—but not always.

“Dirty trick”: Via exactness it often suffices to determine just \( \dim F_k(\Gamma^Q) \).
III. Algorithm: Overview

**Dimension of line bundle sheaf cohomology**

\[
\dim H^i(X; \mathcal{O}_X(D)) = \sum_{Q} h_i(Q) \cdot N_D(Q)
\]

- # (suitable monomials \( R^Q \)) sum ranges over square-free monomials from unions of SR generators

1. Determine all monomials \( Q \) from unions of SR gens.
2. For each such \( Q \) compute the corresponding numbers of SR gen. combinations \( F_k(\Gamma^Q) \)
3. From those determine the multiplicity factors \( h_i(Q) \)
4. For each \( Q \) where \( h_i(Q) \neq 0 \) count the number of rational functions \( N_D(Q) \).
5. Sum over all relevant contributions \( h_i(Q) \cdot N_D(Q) \).

\[ \rightarrow \text{completely algorithmic} \]
Part 4

- Applications
- Cohomology of subvarieties
- Tangent bundle of complete intersections
- Finite group actions (orbifolds & orientifolds)

[Blumenhagen-Jurke-Rahn-Roschy '10]
From the cohomology of a toric ambient space one can descent to the cohomology of a hypersurface, e.g. a Calabi-Yau hypersurface like $\mathbb{P}^4[5]$.

**The Koszul sequence**

Let $S$ be a hypersurface (i.e. divisor) in a toric variety $X$ and $T$ be an arbitrary second divisor of $X$.

\[
0 \longrightarrow \mathcal{O}_X(T - S) \longrightarrow \mathcal{O}_X(T) \longrightarrow \mathcal{O}_S(T) \longrightarrow 0.
\]

- line bundles on ambient space $X$
- line bundle on hypersurface $S \subset X$

$\implies$ Compute $H^i(S; \mathcal{O}_S(T))$ via induced long exact cohomology sequence.

The complete intersection $S = S_1 \cap \cdots \cap S_t$ of several hypersurfaces $S_i \subset X$ can be handled by iteration.
IV. Applications: **Tangent bundle of hypersurfaces**

On a complete intersection $S = S_1 \cap \cdots \cap S_t$ the tangent bundle $T_S$ can be described via two short exact sequences:

**The split Euler sequence**

\[
0 \longrightarrow \mathcal{O}_S^\oplus r \longrightarrow \bigoplus_{k=1}^n \mathcal{O}_S(D_k) \longrightarrow \mathcal{E}_S \longrightarrow 0
\]

\[
0 \longrightarrow T_S \longrightarrow \mathcal{E}_S \longrightarrow \bigoplus_{i=1}^t \mathcal{O}_S(S_i) \longrightarrow 0
\]

Use the Koszul sequence to compute the cohomology dimensions $h^i(S; \mathcal{O}_S(T))$ from the cohomology of $X$.

→ Laborious, but in principle “straightforward”.

Benjamin Jurke (MPI für Physik)
In orbifold or orientifold constructions one has a discrete finite symmetry acting on the space-time and considers the quotient space.

Consider the invariant part of the cohomology.

**Equivariant structure on line bundles**

Let $G$ be a finite group acting holomorphically on $X$. The group element action $g : X \rightarrow X$ on the base space may be lifted to the bundle mapping $\phi_g : L \rightarrow L$. If

$$\phi_g \circ \phi_h = \phi_{gh}$$

this defines an equivariant structure.

This gives a splitting of the cohomology classes:

$$H^i(X; \mathcal{O}_X(D)) = H^i_{\text{inv}}(X; \mathcal{O}_X(D)) \oplus H^i_{\text{non-inv}}(X; \mathcal{O}_X(D)).$$

[Cvetic-García-Etxebarria-Halyerson '10]
The dimensions of those splittings can be computed by applying the uplifted $G$-action on the monomials counted in $\mathcal{N}_D(Q) = \mathcal{N}_{\text{inv}} \oplus \mathcal{N}_{\text{non-inv}}$.

Note that the multiplicity factors $h_i(Q)$ remain unchanged!

$\Rightarrow$ Rather simple to compute!
Consider the following $\mathbb{Z}_3$-action on $\mathbb{P}^2$:

$$g_1 : (x_1, x_2, x_3) \mapsto (\alpha x_1, \alpha^2 x_2, x_3) \quad \text{for} \quad \alpha := \sqrt[3]{1} = e^{\frac{2\pi i}{3}}.$$

Due to the projective equivalence $(x_1, x_2, x_3) \sim (\lambda x_1, \lambda x_2, \lambda x_3)$ between the homogeneous coordinates $x_i$ this base involution is equivalent to

$$g_2 : (x_1, x_2, x_3) \mapsto (x_1, \alpha x_2, \alpha^2 x_3)$$

$$g_3 : (x_1, x_2, x_3) \mapsto (\alpha^2 x_1, x_2, \alpha x_3),$$

The same coordinates form the monomials $R^Q$ used in the algorithm.

Involution mapping can be applied to the monomials.

Choose $g_1$ to be the equivariant structure. For fixpoint-free actions (yielding smooth quotients) all equivariant structures are equivalent.
IV. Applications: Example: $\mathbb{CP}^2/\mathbb{Z}_3$

Apply action to monomials that contribute to the cohomology and read off the corresponding parts by their phases.

Example: Monomials for $\mathcal{O}(-6)$ using $g_1$, i.e. $(x_1, \alpha x_2, \alpha^2 x_3)$:

\[
\begin{align*}
\frac{1}{\alpha^4 u_1^4 \alpha^2 u_2 u_3}, & \quad g_1 \rightarrow 1 \\
\frac{1}{\alpha u_1 \alpha^8 u_2^4 u_3}, & \quad g_1 \rightarrow 1 \\
\frac{1}{\alpha^2 u_1^2 \alpha^6 u_2^6 u_3}, & \quad g_1 \rightarrow \alpha^2 \\
\frac{1}{\alpha u_1 \alpha^6 u_2^3 u_3}, & \quad g_1 \rightarrow \alpha \\
\frac{1}{\alpha^2 u_1^2 \alpha^2 u_2^2 u_3}, & \quad g_1 \rightarrow \alpha \\
\frac{1}{\alpha^3 u_1^3 \alpha^4 u_2^2 u_3}, & \quad g_1 \rightarrow \alpha^2 \\
\frac{1}{\alpha u_1 \alpha^4 u_2^2 u_3}, & \quad g_1 \rightarrow \alpha \\
\frac{1}{\alpha^2 u_1 \alpha^4 u_2^2 u_3}, & \quad g_1 \rightarrow 1
\end{align*}
\]

Result:

\[
h^2(\mathbb{P}^2; \mathcal{O}(-6)) = (4_{\text{inv}}, 3\alpha, 3\alpha^2) \quad \rightarrow \quad h^2(\mathbb{P}^2/\mathbb{Z}_3; \mathcal{O}(-6)) = 4
\]
Now: Make it simple & easy to use!

Part 5

- Algorithm implementation cohomCalc
- Outlook
V. Implementation: **cohomCalg**

**cohomCalg**

- fast sheaf cohomology computation for line bundles on toric varieties

**Koszul extension**

→ **Google** for **cohomCalg**

or try the core algorithm online:

→ **cohomcalg.benjaminjurke.net**

**cohomCalg**

- high-speed, cross-platform
- C++ implementation **cohomCalg**
  - Windows / Mac / Linux
  - open source, GLPv3
  - multi-core support

**cohomCalg Koszul extension**

- Mathematica interface
  - Hypersurfaces & complete intersections
  - (co-)tangent bundle, $\Lambda^2 T^* S$
  - Hodge diamond
  - Monads
Example: Line bundles on toric variety $dP_3$

The C++ core program takes care of the actual algorithm that computes line bundles on toric spaces.
V. Implementation: Tangent bundle cohomology of hypersurface

Example: The resolved \( \mathbb{P}_{11222}^8 \)

```
In[181]:= P11222BlowUp = {
    (*Coordinates*){v1, v2, v3, v4, v5, vX},
    (*Stanley Reisner*){{v1, v2}, {v1, v2}},
    (*Equivalence Relations*){{1, 0}, {1, 0}, {2, 1}, {2, 1}, {2, 1}, {0, 1}}
};
CalabiYauHyperSurface = {{8, 4}};
CohomologyOf["TangentBundle", P11222BlowUp, CalabiYauHyperSurface, "Calabi-Yau", "Verbose2"]
```

<table>
<thead>
<tr>
<th>( O_S^{\oplus} \oplus \bigoplus_{k=1}^n [O_S[D_k]] )</th>
<th>( E_S )</th>
<th>( T_S )</th>
<th>( E_S )</th>
<th>( \bigoplus_{i=1}^l [O_S[S_i]] )</th>
<th>( T_S )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( O_S[1,0] \oplus O_S[2,1] \oplus O_S[0,1] )</td>
<td>2 ( A_{21} )</td>
<td>21</td>
<td>0</td>
<td>21</td>
<td>104</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>3</td>
<td>21</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Calculation of TangentBundle cohomology done:
-----------------------------------------------
Total number of line bundles: 42
Newly computed: 18
Total time needed: 0.593 seconds
V. Implementation: Hodge diamond of a CICY 4-fold

CICY 4-fold in the context of F-Theory GUT Vacua

\[
\begin{align*}
\text{Example4Fold} &= \{ \\
&\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}, \\
&\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}, \\
&\{v_1, v_2, v_3, v_4, v_5, v_6, v_10\}, \\
&\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}, \\
&\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}, \\
&\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}\}, \\
\text{CompleteIntersection} &= \{\{6, 6, 6, 0\}, \{0, 0, 2, 1, 1\}\}; \\
\text{CohomologyOf} &= \{\text{"HodgeDiamond"}, \text{Example4Fold}, \text{CompleteIntersection}, \text{"Calabi-Yau"}\}
\end{align*}
\]

Calculation of HodgeDiamond cohomology done:

Total number of line bundles: 478
Newly computed: 264
Total time needed: 31.233 seconds

\[\chi = 6768\]

[Grimm-Krause-Weigand ’09]

The Mathematica frontend provides convenient functionality, utilizing the previously discussed methods.
Conclusions

Presented material

- An easy and efficient way to compute line bundle-valued cohomology group dimensions of a toric variety.

- The algorithm implementation `cohomCalc`.

- Methods for the computation of various vector bundles on toric subspaces with a convenient Mathematica frontend.

- An extension of the proposed algorithm to compute quotient space cohomology of a toric variety, i.e. methods to calculate line bundle cohomology on orbifolds and orientifolds.
Related work in progress & ideas

- Exploration of target space duality in \((0, 2)\) het. models.
  [Blumenhagen-Rahn '11 (upcoming)]

- Construction and analysis of new Calabi-Yau 3-folds.
  [Jurke-Rahn '11 (upcoming)]
Outlook

Long term applications

- Scans over extremely large string landscape sets and analysis of various phenomenological properties...

➞ Cloud computing!

The end
Long term applications

- Scans over extremely large string landscape sets and analysis of various phenomenological properties...

→ Cloud computing!

The end

...well, not quite yet!
Part 6

- What is “the Cloud”?  
- How can it be utilized for science?
VI. Cloud Computing: Performance vs. Costs

→ Over time: “More bang for the buck!”
VI. Cloud Computing: What is “the Cloud”?

PDP-10 ➔ PC ➔ ?

- 1966: PDP-10 — First computer that made time-sharing common!
  ➔ Small terminals accessing huge mainframe
- 1981: IBM PC — First computer for personal / private home use!
  ➔ Small independent versatile powerhouses
- Late 2000s: More and more services remotely accessed.
  ➔ A step back?
The improvements in computer speed allow to raise the level of abstraction by a huge margin. Classical approaches like time sharing are replaced by true virtualization.

**Virtualization**

A virtual machine simulates all relevant structures of a computer. From a programmers perspective one effectively operates on an entirely separate computer.

- One no longer has to care about the hardware details of the machine!

On modern machines the performance losses of virtualization for most applications are no longer significant.

- True detachment of hardware and software!
VI. Cloud Computing: Remote (Super-)Computing — Pros

Using virtualization one can effectively rent a remote computer and for all practical purposes operate on it similar to a local machine.

- **Maintenance**: One big centralized computer cluster is much easier to maintain than thousands of smaller systems.
- **Load spikes & uptime**: Since a virtual machine is not bound to specific hardware, one can easily move it to a different machine or (dynamically) associate more hardware to it.
- **Costs**: Due to dynamic allocation of resources, the actual hardware is more efficiently used.
However, there are also new challenges to be faced:

- **Restrictions**: The virtualization may be limited to certain specific (virtual) operating systems, which makes it challenging (or impossible) to use existing software.

- **Security**: The data is no longer “at home”, privacy questions arise.

Do those limitations largely affect *scientific* computing?
NSF grant to B. Nelson, J. Gray, Y.-H. He and V. Jejjala to utilize the **Microsoft Azure cloud platform** for computational problems relevant for string theory.