
LA COLUMNA DE MATEMÁTICA COMPUTACIONAL

Sección a cargo de

Tomás Recio Muñiz

El objetivo de esta columna es presentar de manera sucinta, en cada uno de los números de La Gaceta, alguna cuestión matemática en la que los cálculos, en un sentido muy amplio, tengan un papel destacado. Para cumplir este objetivo el editor de la columna (sin otros méritos que su interés y sin otros recursos que su mejor voluntad) quisiera contar con la colaboración de los lectores, a los que anima a remitirle (a la dirección que se indica al pie de página¹) los trabajos y sugerencias que consideren oportunos.

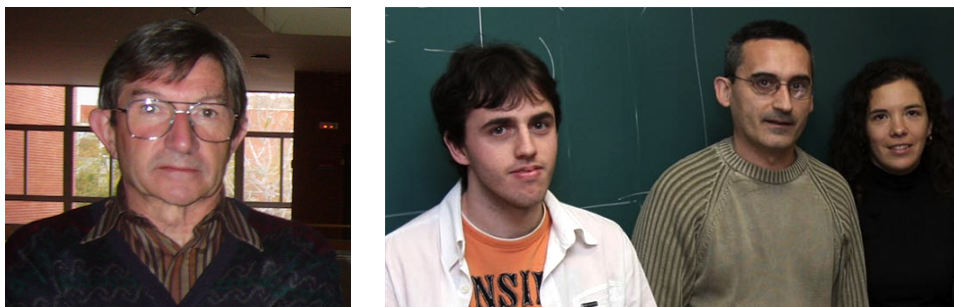
EN ESTE NÚMERO. . .

En este número de *La Gaceta* presentamos un artículo de un grupo de investigadores del equipo *Psycotrip* (Programación y Cálculo Simbólico) de la Universidad de La Rioja, en el que se describe el programa *fKenzo*, para la realización de cálculos en Topología Algebraica. Se trata de una traducción al castellano de una versión resumida de un artículo que los autores publicarán en breve en el *Journal of Symbolic Computation* (doi:10.1016/j.jsc.2011.01.005, disponible en línea desde el 3 de febrero de 2011 en <http://www.elsevier.com/locate/jsc>), a cuyo editor agradecemos las facilidades concedidas.

Hacia el año 1985, Francis Sergeraert, investigador del Institut Fourier, en Grenoble (Francia), propuso un concepto teórico, el de la Homología Efectiva, que debería permitir resolver muchos problemas de calculabilidad en Topología Algebraica y, además, debería dar lugar a programas reales para el cálculo de invariantes en esa disciplina. Sobre el año 1989 se terminó una primera versión de un conjunto de programas que implementaban las ideas de la Homología Efectiva, dando lugar a un sistema denominado EAT (por «Effective Algebraic Topology»). Esos esfuerzos estuvieron relacionados con la tesis doctoral de Julio Rubio (Zaragoza 1988, Grenoble 1991), dirigida por Sergeraert.

Conforme el desarrollo de estos avances fue consolidándose, se reconoció que, más allá del carácter «efectivo» de las técnicas (es decir, de que se podían ejecutar en un ordenador), se encontraba una revisión de la misma Topología Algebraica desde el punto de vista de las matemáticas constructivas, por lo que Sergeraert renombró esta línea de trabajo como CAT (por «Constructive Algebraic Topology»). Como

¹Tomás Recio, Departamento de Matemáticas, Facultad de Ciencias, Universidad de Cantabria, 39071-Santander; tomas.recio@unican.es



A la izquierda, el profesor Francis Sergeraert. A la derecha, Jónathan Heras, Julio Rubio y Vico Pascual.

una de las gatas de Sergeraert en ese momento se llamaba *Kenzo*, ese fue el nombre elegido para la siguiente versión de EAT que, con mejoras y modificaciones, es la que pervive hasta la fecha. En paralelo con la evolución de *Kenzo*, que siguió siendo dirigida por Sergeraert, en la Universidad de La Rioja se lanzó un proyecto para estudiar *Kenzo* formalmente, como sistema software.

El primer resultado destacable en esa nueva línea fue la tesis de Vico Pascual en el año 2002, codirigida por Laureano Lambán y Julio Rubio. Conforme la comprensión de *Kenzo* avanzaba, desde el punto de vista de la Ingeniería del Software, se fue haciendo posible el emprender la realización de una interfaz de usuario amigable para el sistema.

El resultado de esa iniciativa es el sistema *fKenzo* que se presenta en este artículo, que es una parte de la tesis doctoral de Jónathan Heras, codirigida por Vico Pascual y Julio Rubio.

En la actualidad el profesor Francis Sergeraert es «Profesor Honoraire» del Institut Fourier, desde su retiro de las labores docentes en el año 2005, y es también miembro honorario e investigador oficial y realmente activo en los proyectos del grupo *Psycotrip* que dirige el profesor Julio Rubio, y en el que colaboran los otros dos autores de este artículo.

fKenzo: una interfaz de usuario para realizar cálculos en Topología Algebraica[†]

por

Jónathan Heras, Vico Pascual, Julio Rubio y Francis Sergeraert

RESUMEN. *fKenzo* (del inglés *friendly Kenzo*) es una interfaz gráfica de usuario que proporciona un acceso cómodo al sistema *Kenzo*, un programa *Common Lisp* dedicado a la Topología Algebraica. El sistema *fKenzo* está compuesto por la componente gráfica en sí, por una capa intermedia basada en XML que conecta la interfaz con *Kenzo*, y por el propio sistema *Kenzo*. En este artículo se describen las principales características de *fKenzo*, así como sus ventajas y limitaciones con respecto a *Kenzo*.

INTRODUCCIÓN

La Topología Algebraica es una rama de las Matemáticas que estudia los espacios topológicos por medios algebraicos, en particular a través de *invariantes algebraicos*, como los grupos de homología y homotopía. Por ejemplo, dos espacios que tienen grupos de homología diferentes (no isomorfos) tienen tipos de homotopía distintos.¹ La evolución moderna de la Topología Algebraica se centra en problemas *estructurales*, dejando a un lado el interés *computacional*. Una posible razón para dicha evolución reside en que, en general, el cálculo de grupos de homología u homotopía de espacios arbitrarios es una tarea compleja, fuera del alcance de un topólogo que «sólo» trabaje con lápiz y papel.

El fascinante ejemplo del Álgebra Conmutativa, donde los especialistas utilizan, desde hace ya tiempo, poderosas herramientas computacionales tales como *Macaulay*, *Cocoa* o *Singular* para ayudarse en su trabajo, produciendo nuevos problemas *matemáticos*, no tiene su equivalente en Topología Algebraica. ¿Por qué? Un tema interesante para los investigadores en Historia de las Matemáticas. Una posible razón es la siguiente: los programas para Topología Algebraica requieren un alto nivel de *programación funcional*, pero ese no es el tema de este trabajo.

Este artículo está dedicado a presentar el sistema *fKenzo* (del inglés *friendly Kenzo*), una interfaz gráfica que permite a un topólogo el uso del programa *Kenzo* para aplicaciones simples, sin necesidad de preocuparse por las cuestiones técnicas que son inevitables cuando queremos utilizar toda la potencia del sistema *Kenzo*.

[†]Parcialmente financiado por el proyecto MTM2009-13842-C02-01.

¹Desafortunadamente (o afortunadamente, dependiendo del punto de vista) el recíproco no es cierto.

Para ilustrar la capacidad computacional de *fKenzo*, vamos a considerar el escenario hipotético de un curso de doctorado dedicado a *fibraciones* que, en particular, introduce los funtores *espacio de lazos* Ω y *espacio clasificante* B . En el contexto simplicial, el libro [7] es una buena referencia para estos temas. En este entorno, si X es un *espacio* conexo, su espacio de lazos ΩX es un *grupo* simplicial (ΩX es el espacio constituido por las funciones continuas de la circunferencia S^1 en X que preservan el punto base), el grupo estructural de una fibración universal $\Omega X \hookrightarrow PX \rightarrow X$. Recíprocamente, si G es un *grupo* simplicial, el espacio clasificante BG es el espacio base de una fibración universal $G \hookrightarrow EG \rightarrow BG$. Esta simetría entre ambas situaciones nos lleva de manera natural a la siguiente cuestión: en un contexto apropiado, ¿son los funtores Ω y B inversos el uno del otro? En el caso de la composición $B\Omega$, podemos utilizar *fKenzo* para comparar los primeros grupos de homología de la 2-esfera S^2 con los de ΩS^2 y los de $B\Omega S^2$, y dar cierta credibilidad a la relación $B\Omega = \text{id}$ en la categoría homotópica. El grupo simplicial ΩS^2 puede a su vez ser usado para comparar del mismo modo ΩS^2 y $\Omega B\Omega S^2$, llegando a la conclusión $\Omega B = \text{id}$.

Un (buen) estudiante se puede preguntar por qué no hemos considerado en primer lugar el caso más simple de la circunferencia, la 1-esfera S^1 . Si usásemos de nuevo *fKenzo*, esta vez no obtendríamos el resultado esperado, a pesar de que la afirmación $B\Omega S^1 \sim S^1$ es cierta. Sin embargo, en este caso no se puede usar la sucesión espectral de Eilenberg-Moore para calcular $H_*\Omega S^1$, debido a que S^1 no es simplemente conexo y *fKenzo* comprueba este hecho. La comparación simétrica entre S^1 y $\Omega B S^1$ también falla, pero por otra razón: la versión «estándar» de S^1 como espacio topológico es un grupo, pero a la presentación simplicial estándar de S^1 no se le puede dotar de una estructura de grupo *simplicial*. Esta es una buena oportunidad para introducir el espacio de Eilenberg-MacLane $K(\mathbb{Z}, 1)$, que es el modelo de Kan «minimal» para la circunferencia S^1 y a su vez un grupo simplicial. Ahora sí, si comparamos en *fKenzo* los primeros grupos de homología de $K(\mathbb{Z}, 1)$ con los de $\Omega BK(\mathbb{Z}, 1)$ obtenemos el resultado esperado. En general podemos decir que las equivalencias homotópicas $B\Omega X \sim X$ y $\Omega BG \sim G$ son ciertas si X es un espacio conexo y G es un grupo simplicial, ver [7].

Creemos que esto ilustra el modo en que se puede utilizar *fKenzo* como una herramienta de ayuda para la Topología Algebraica. Merece la pena resaltar que todos los espacios de los ejemplos anteriores (exceptuando las esferas S^1 y S^2) no son conjuntos simpliciales de tipo finito. Por lo tanto, el cálculo de sus grupos de homología, sin la herramienta apropiada, es una tarea que supone un desafío para las capacidades de un principiante en Topología Algebraica. Afortunadamente, *Kenzo* es capaz de calcular de manera rápida estos grupos gracias a la teoría de la *Homología Efectiva*. Esto proporciona un aspecto experimental a la Topología Algebraica, aspecto que el Álgebra Conmutativa heredó ya hace bastantes años de los sistemas de cálculo simbólico.

Aunque este artículo no está dedicado a introducir los conceptos y técnicas de la Homología Efectiva (que pueden encontrarse detallados por ejemplo en [9]), vamos a intentar dar una pequeña motivación en este párrafo, tomando como ejemplo la homología de los espacios de lazos. Hablando informalmente, las técnicas de Homología Efectiva proporcionan una conexión entre un objeto de tipo no finito X y un

objeto finito Y que tiene los mismos grupos de homología que X ; por lo tanto, el problema de calcular los grupos de homología de X se reduce a calcular los grupos de homología de Y . En el ejemplo, partiríamos de un objeto con homología efectiva como la esfera de dimensión tres S^3 (tiene homología efectiva porque su complejo de cadenas simplicial es de tipo finito). A continuación le aplicaríamos el functor espacio de lazos Ω (recuérdese, se trata del espacio de aplicaciones continuas de la circunferencia S^1 en S^3). El modelo simplicial para ese espacio ΩS^3 es un conjunto con un número infinito de símplices (se trata, esencialmente, de un grupo libre no abeliano generado por símplices de S^3 ; ver la construcción G de Kan en [7]). El complejo de cadenas asociado a ΩS^3 es por tanto el objeto X en nuestra explicación anterior, que no es de tipo finito. Las técnicas de la homología efectiva permiten construir algorítmicamente una equivalencia de homotopía con un objeto de tipo finito Y (en el ejemplo de ΩS^3 se trata de una variante de la construcción *Cobar* de Adams; ver [9]). De ese modo, ΩS^3 pasa a ser un objeto con homología efectiva y, desde el punto de vista de *Kenzo* o *fKenzo*, no hay ninguna diferencia estructural entre él y el espacio más simple S^3 . La consecuencia de este hecho va más allá de que podamos calcular los primeros grupos de homología de ΩS^3 ; además, podemos volverle a aplicar todas las construcciones que tenemos a nuestra disposición en *Kenzo*. En particular, podemos volver a tomar su espacio de lazos $\Omega(\Omega S^3)$ (habitualmente denotado $\Omega^2 S^3$), lo que da una solución elegante para el cálculo de la homología de los espacios de lazos iterados.

El resto del artículo está organizado del siguiente modo. En la primera sección mostramos cómo serían procesados en *Kenzo* los ejemplos sobre los espacios de lazos y espacios clasificantes que hemos presentado en esta introducción. En la sección 2, se consideran los mismos ejemplos utilizando esta vez la interfaz *fKenzo*. Una descripción de las capacidades de *fKenzo* es el tema de la sección 3. En la última sección, comparamos las ventajas y desventajas de *Kenzo* con respecto a *fKenzo*. El artículo termina con una sección de conclusiones y posibles líneas de trabajo futuro, y, finalmente, incluimos la bibliografía.

1. *Kenzo* ANTES DE *fKenzo*

El programa *Kenzo* es una colección de ficheros *Common Lisp* que debe ser usado en un entorno *Common Lisp*. La página web de *Kenzo* [3] proporciona la información necesaria para que un topólogo pueda instalar un entorno *Common Lisp* en su ordenador,² y a continuación trabajar con *Kenzo*. Este programa está constituido por una colección de clases (complejos de cadenas, conjuntos simpliciales, grupos simpliciales, etcétera) y un buen número de funciones que permiten a un usuario la gestión de los objetos tradicionales de la Topología Algebraica, y en particular el cálculo de algunos grupos de homología y homotopía.

Una sesión de *Kenzo* en la cual intentamos comparar la 2-esfera S^2 y el espacio clasificante $B\Omega S^2$ podría comenzar del siguiente modo:

²Algunos entornos *Common Lisp* son libres y gratuitos.

```

> (setf S2 (sphere 2)) ✘
[K1 Simplicial-Set]

```

Esta interacción con *Kenzo* hay que interpretarla del siguiente modo. El símbolo inicial '>' es el carácter que se muestra en nuestro entorno *Common Lisp* para indicar que el sistema está a la espera de órdenes. El usuario escribe una instrucción *Lisp*, en este caso `(setf S2 (sphere 2))`, y la cruz de malta ✘ (que no es visible en la pantalla del usuario) indica en este texto el final de la instrucción *Lisp*. La tecla de retorno pide a *Lisp* que *evalúe* la instrucción. La 2-esfera S^2 es construida por medio de la función `sphere` de *Kenzo*, que toma el valor 2 como argumento, y es asignada al símbolo S2 para ser usada en el futuro. Al evaluar una instrucción, *Lisp devuelve* un objeto, que es el resultado de la evaluación. En el ejemplo, es el objeto *Kenzo* número #1, un Simplicial-Set, objeto *Lisp* que implementa a la 2-esfera, mostrado como [K1 Simplicial-Set]. La estructura interna de dicho objeto está compuesta de un conjunto de ricas estructuras de datos, en concreto varias componentes funcionales, pero no es mostrada al usuario.

Kenzo también puede construir el espacio de lazos ΩS^2 , un grupo simplicial, y después el espacio clasificante de este grupo, $B\Omega S^2$, que sólo es un conjunto simplicial, dado que el grupo ΩS^2 no es abeliano:

```

> (setf OS2 (loop-space S2)) ✘
[K6 Simplicial-Group]
> (setf BOS2 (classifying-space OS2)) ✘
[K18 Simplicial-Set]

```

Es un resultado conocido que, para $p \geq 0$, se cumple $H_p \Omega S^2 = \mathbb{Z}$. Para valores pequeños de p , *Kenzo* puede calcular dichos grupos sin ningún problema, utilizando la *versión efectiva* de la sucesión espectral de Eilenberg-Moore aplicada al conjunto simplicial S2. Por ejemplo:

```

> (homology OS2 6) ✘
Homology in dimension 6 :
Component Z

```

que se debe interpretar como $H_6 \Omega S^2 = \mathbb{Z}$. Así es posible comparar algunos grupos de homología de S^2 y $B\Omega S^2$:

```

> (homology S2 2) ✘
Homology in dimension 2 :
Component Z
> (homology BOS2 2) ✘
Homology in dimension 2 :
Component Z
> (homology S2 6) ✘
Homology in dimension 6 :
Component Z
> (homology BOS2 6) ✘
Homology in dimension 6 :
Component Z

```

Después de Homology in dimension 6 no se muestra ninguna *componente*, lo que significa que el correspondiente grupo de homología es nulo. *Kenzo* también

informa al usuario de que $H_p S^2 = H_p B\Omega S^2$ para $p = 2$ y $p = 6$. Se obtiene el mismo resultado para otros valores de p , lo que implica que merece la pena estudiar la cuestión de la equivalencia homotópica $S^2 \overset{?}{\sim} B\Omega S^2$.³

La cuestión simétrica $G \overset{?}{\sim} \Omega BG$ requiere un *grupo* G , por ejemplo $G = \Omega S^2$. El lector puede ahora entender el objetivo de las siguientes instrucciones *Kenzo*:

```
.....
> (setf OBOS2 (loop-space BOS2)) ✘
[K252 Simplicial-Group]
> (homology OBOS2 6) ✘
Homology in dimension 6 :
Component Z
.....
```

De nuevo el usuario puede comprobar que $H_p \Omega S^2 = H_p \Omega B\Omega S^2$ para valores pequeños de p . Los mismos cálculos se pueden repetir con otros espacios iniciales, sugiriendo claramente la cuestión general de la equivalencia homotópica $G \overset{?}{\sim} \Omega BG$.

Intentemos realizar el mismo trabajo con el grupo $G = S^1$. Dado que la circunferencia S^1 es un subgrupo del grupo multiplicativo \mathbb{C}^* , se puede definir el espacio clasificante BS^1 :

```
.....
> (setf S1 (sphere 1)) ✘
[K395 Simplicial-Set]
> (setf BS1 (classifying-space S1)) ✘
Error: No methods applicable for generic function
#<STANDARD-GENERIC-FUNCTION CLASSIFYING-SPACE>
with args ([K395 Simplicial-Set]) of classes (SIMPLICIAL-SET)
.....
```

Sin embargo la implementación por defecto de *Kenzo* para la circunferencia S^1 es un *conjunto* simplicial con un vértice y una arista. Como se muestra claramente, este objeto tiene la estructura (*clase* en la jerga de computadores) de *conjunto* simplicial, así que la función genérica `classifying-space` no se puede aplicar a dicho objeto. La Topología Algebraica Computacional debe usar modelos *combinatorios*, y al modelo simplicial más sencillo de la circunferencia no se le puede dotar de una estructura de *grupo* simplicial: un grupo simplicial es necesariamente un conjunto simplicial de Kan, ver [7], y el modelo minimal de Kan para la circunferencia es el modelo simplicial estándar del espacio de Eilenberg-MacLane $K(\mathbb{Z}, 1)$; un conjunto simplicial *que no es de tipo finito* pero que sin embargo se puede construir y usar en el sistema *Kenzo*:

```
.....
> (setf KZ1 (k-z 1)) ✘
[K513 Abelian-Simplicial-Group]
> (setf BKZ1 (classifying-space kz1)) ✘
[K525 Abelian-Simplicial-Group]
> (setf OBKZ1 (loop-space bkz1)) ✘
[K537 Simplicial-Group]
.....
```

Los grupos de homología $H_p \Omega BK(\mathbb{Z}, 1)$ pueden ahora ser calculados y comparados con los conocidos grupos de homología de la circunferencia:

³*Kenzo* permite al usuario calcular, por ejemplo, cada grupo de homología $H_p B\Omega S^2$ con $p \leq 8$, de un modo más adecuado, pero en esta breve nota hemos preferido evitar la salida de resultados excesivamente verbosa producida por *Kenzo*.

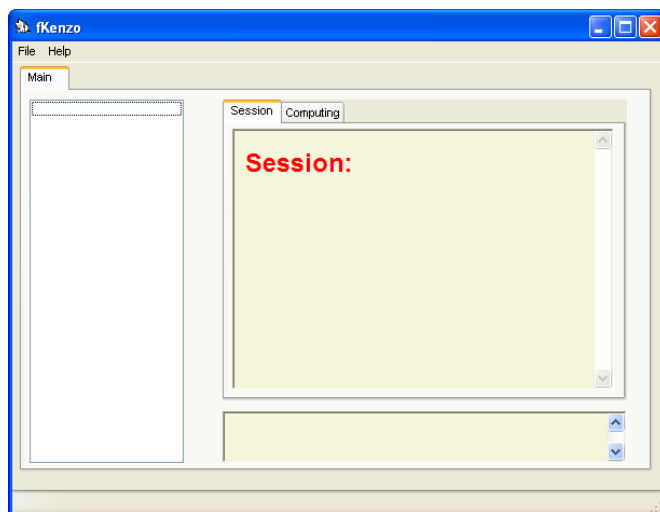


Figura 1: Ventana inicial de *fKenzo*.

```

> (homology OBKZ1 1) ✘
Homology in dimension 1 :
Component Z
> (homology OBKZ1 6) ✘
Homology in dimension 6 :

>

```

2. EL MISMO EJEMPLO CON *fKenzo*

Para repetir los mismos cálculos con *fKenzo*, el primer paso consiste en descargarse el instalador desde la página web indicada en [4]; por el momento *fKenzo* sólo se encuentra disponible para Windows. Después del proceso de instalación (no es necesario instalar ningún programa adicional), el usuario puede pulsar sobre el icono de *fKenzo*, accediendo a la interfaz vacía mostrada en la figura 1.

Dado que estamos planeando trabajar con conjuntos simpliciales y grupos simpliciales, podemos usar *File* → *Add Module* y ahí seleccionar el fichero *Simplicial-Groups.omdoc*. En ese momento la interfaz cambia, incluyendo dos nuevos menús llamados *Simplicial-Sets* y *Simplicial-Groups*. Al desplegar estos menús encontramos distintas opciones que permiten construir, en particular, esferas, espacios de lazos o espacios clasificantes. Al seleccionar la opción *Simplicial-Sets* → *sphere*, *fKenzo* pide un número natural limitado, por defecto en *Kenzo*, a 14. El espacio denotado por *SS 1* aparece en el lado izquierdo de la pantalla; al seleccionarlo (ver figura 2), en la parte inferior derecha de la pantalla aparece la notación matemática estándar del espacio. En la parte superior derecha del panel se muestra el historial de los espacios construidos.

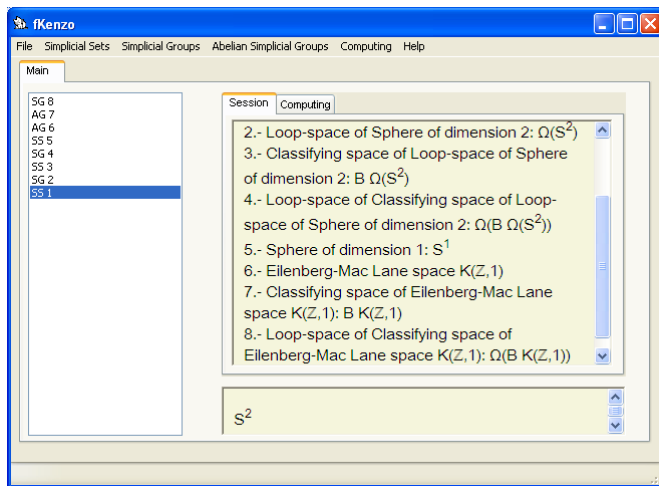


Figura 2: Ejemplo de sesión.

Al intentar construir un espacio clasificante desde el menú *Simplicial-Groups*, *fKenzo* informa de la necesidad de construir un grupo simplicial (de este modo se evita un posible error). Se puede entonces construir el espacio ΩS^2 y, dado que ΩS^2 es el único grupo simplicial en la sesión, al seleccionar la opción *classifying-space*, ΩS^2 es el único espacio disponible en la lista mostrada por *fKenzo*. Para trabajar con los espacios de Eilenberg-MacLane, se debe cargar el módulo de *Abelian-Simplicial-Groups*. De este modo todos los espacios mencionados en la sección anterior se pueden construir en *fKenzo*; ver figura 2.

Al cargar el fichero *Computing.omdoc*, aparece el menú *Computing*, en el que se puede seleccionar la opción *Homology*. De este modo se reproducen los cálculos realizados en la sección 1, como puede observarse en la figura 3.

Si el usuario intenta realizar algún cálculo para el que la teoría de la homología efectiva no prevé ningún algoritmo, como en el caso de $H_4(B\Omega^2 S^2)$, *fKenzo* le informa de que no puede calcular $H_4(B\Omega^2 S^2)$ (ver figura 4) debido, en este caso concreto, a un problema con el grado de conexión del espacio. Por lo tanto, *fKenzo* no sólo permite una interacción más cómoda que *Kenzo*, sino que también guía al usuario, evitando errores de ejecución (ver la idea de *capa intermedia* en el artículo [5]).

3. *fKenzo*: DESCRIPCIÓN DEL SISTEMA

fKenzo es una interfaz de usuario extensible. Basa su organización en el uso de algunas herramientas *OpenMath*, como los diccionarios de contenido y los documentos *OpenMath*; ver [2]. Las estructuras de datos que utiliza el sistema y que codifican estructuras matemáticas (esferas, espacios clasificantes, espacios de lazos, etcétera) están organizadas en diccionarios de contenido considerando las estructuras matemáticas que representan. Esto define una estructura modular para *fKenzo*.

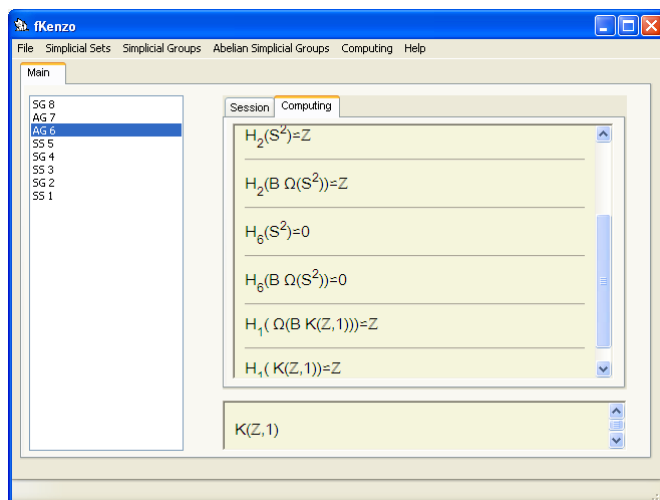


Figura 3: Ejemplo de cálculos.

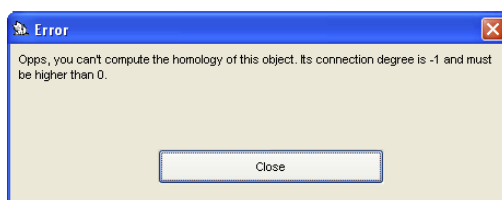


Figura 4: Error del grado de reducción.

Cuando en el menú *File* seleccionamos la opción *Add Module*, se nos muestra una lista de los documentos *OpenMath* disponibles. El efecto de añadir un nuevo módulo es que la interfaz del usuario cambia, normalmente incluyendo un nuevo menú con las funcionalidades correspondientes al documento *OpenMath*.

En su distribución actual, *fKenzo* cuenta con cinco módulos: *Chain-Complexes*, *Simplicial-Sets*, *Simplicial-Groups*, *Abelian-Simplicial-Groups* y *Computing*. Además, se pueden descargar algunos módulos *experimentales*, a través del gestor de *plug-ins* que se encuentra en el menú *Help*, dando la posibilidad de integrar *fKenzo* con el sistema de cálculo simbólico *GAP* o el demostrador de teoremas *ACL2*.⁴

El módulo *Simplicial-Set* contiene la mayor parte de la funcionalidad de *Kenzo*. Incluye opciones para construir espacios desde cero: esferas, espacios de Moore, conjuntos simpliciales finitos (utilizando la sintaxis descrita en [3]), etcétera. La op-

⁴Estos aspectos experimentales son una de las razones para usar *OpenMath* como lenguaje de especificación, ya que permite comunicarnos con sistemas de cálculo simbólico, como *GAP*, y también incluir información axiomática que puede ser utilizada para probar propiedades de programas con *ACL2*. La otra razón para usar *OpenMath* es que, combinado con una hoja de estilos, nos va a permitir mostrar en nuestra aplicación las expresiones matemáticas con su notación estándar.

ción de conjuntos simpliciales finitos permite definir complejos simpliciales finitos (además, un módulo especializado en complejos simpliciales está disponible como una extensión del sistema a través del menú *Help*). El módulo *Simplicial-Set* también contiene operaciones para definir espacios a partir de otros previamente construidos: productos cartesianos, suspensiones, etcétera. El módulo *Simplicial-Groups* contiene los operadores para definir espacios de lazos y espacios clasificantes, y además carga automáticamente el módulo *Simplicial-Set* —si no estaba previamente en memoria—, pues dicho módulo es necesario para construir espacios de lazos. El módulo *Abelian-Simplicial-Groups* contiene sólo una opción (espacios de Eilenberg-MacLane). El módulo *Chain-Complexes* incluye algunas operaciones definidas en el nivel algebraico pero no en el simplicial, como el producto tensorial.

Además de estos módulos estructurales, existe un módulo especial llamado *Computing*, que permite al usuario calcular grupos de homología y homotopía de los espacios construidos. Es necesario entender que los algoritmos para el cálculo de la homología y de la homotopía son muy diferentes. Los (primeros) grupos de homología de los espacios construidos con *Kenzo/fKenzo* pueden ser calculados gracias a la teoría de la homología efectiva, ver [9]. Por el contrario, sólo algunos grupos de homotopía pueden calcularse con *fKenzo* (recomendamos al lector interesado que consulte la documentación de *Kenzo* o de *fKenzo* para saber más sobre este tema).

El aspecto visual del panel de *fKenzo* ha sido brevemente descrito en la sección previa. La pestaña «*Main*» contiene, en su lado izquierdo, una lista de los espacios construidos en la sesión actual, identificados por su tipo (CC = *Chain Complex*, SS = *Simplicial Set*, SG = *Simplicial Group*, AG = *Abelian Simplicial Group*) y un número de identificación (este identificador se utiliza internamente para mantener una traza del origen de los espacios, lo que permite la reutilización de resultados intermedios, obteniendo así un rendimiento comparable al de *Kenzo*, que usa esta técnica que ha sido ahora exportada a *fKenzo*). Al seleccionar uno de los espacios de la lista, su notación estándar aparece en la parte inferior derecha. En la parte superior derecha hay dos pestañas: «*Session*» (que contiene una descripción textual de los espacios construidos, ver figura 2) y «*Computing*» (que contiene los grupos de homología y homotopía calculados en la sesión, ver figura 3).

4. *Kenzo* MÁS ALLÁ DE *fKenzo*

La Topología Algebraica es un rama compleja e inmensa de las Matemáticas, en la que se mezclan en particular Álgebra y Topología (combinatoria). Los diseñadores de programas para el cálculo simbólico siempre se encuentran ante el mismo problema para organizar un sistema:

1. Proporcionar un paquete de funciones en el lenguaje de programación L , permitiendo al usuario de ese lenguaje cargar dicho paquete en el entorno estándar de L , y usar sus funciones y procedimientos. Esta es la solución seguida en *Kenzo* con respecto al lenguaje *Common Lisp*. Ventaja: la libertad dada por el lenguaje L se mantiene; desventaja: las dificultades técnicas del lenguaje L también.

2. Proporcionar una interfaz gráfica de usuario con botones, menús y otros elementos usuales. Este es el estilo de *fKenzo*. Ventaja: se le da al usuario inexperto un acceso directo a los cálculos más sencillos sin necesidad de aprender el lenguaje *L*; desventaja: algunas aplicaciones pueden ser difíciles de implementar en una interfaz gráfica, mientras que pueden ser mucho más accesibles programando directamente en el lenguaje *L*.

Damos a continuación dos ejemplos de resultados que se pueden obtener en el entorno original de *Kenzo*, pero que desafortunadamente se encuentran fuera del alcance de *fKenzo* y que parecen difíciles de integrar en cualquier interfaz gráfica.

En el artículo [10] se presenta la siguiente interacción con *Kenzo* que permite obtener unos resultados inesperados. Sea X el espacio proyectivo complejo $P^n\mathbb{C}$. Dicho espacio es un subconjunto del espacio proyectivo complejo infinito $P^\infty\mathbb{C}$, y la clase fundamental de $P^n\mathbb{C}$ es a su vez un generador de $H_{2n}P^\infty\mathbb{C}$. Sucede que $P^\infty\mathbb{C}$ tiene el mismo tipo de homotopía que el espacio de Eilenberg-MacLane $K(\mathbb{Z}, 2)$. Este espacio de Eilenberg-MacLane tiene una versión canónica *minimal* como un conjunto simplicial K_2 , un objeto importante en el cálculo de grupos de homotopía en *Kenzo*. El conjunto simplicial K_2 puede ser construido tanto en *Kenzo* como en *fKenzo*. Este conjunto simplicial es «minimal» pero no es de tipo finito; sin embargo, los métodos de la *homología efectiva* nos permiten calcular la ... *homología efectiva* de K_2 . En particular, el programa *Kenzo* puede producir un representante c , un ciclo, de un generador de $H_{2n}(K_2)$. El ciclo c es una \mathbb{Z} -combinación finita de $2n$ -símplices de K_2 . Utilizando el ciclo c , *Kenzo* nos permite definir $S_c \subset K_2$, el subconjunto simplicial generado por las componentes de c ; no es muy difícil probar entonces que la homología relativa $H_*(K_2, S_c)$ es nula hasta dimensión $2n + 1$, por lo que el conjunto simplicial S_c tiene el mismo tipo de homotopía que $P^n\mathbb{C}$. El programa *Kenzo* puede calcular esta homología relativa, que prueba que S_c es una triangulación de un conjunto simplicial del tipo de homotopía de $P^n\mathbb{C}$. Así es posible obtener, en unos pocos segundos, una triangulación del tipo de homotopía de $P^5\mathbb{C}$ con (1, 0, 5, 40, 271, 1197, 3381, 5985, 6405, 3780, 945) símplices, esto es, 1 vértice, 5 triángulos, 40 tetraedros, ..., 945 10-símplices. En particular, *Kenzo* calcula una triangulación de $P^2\mathbb{C}$ como conjunto simplicial con (1, 0, 2, 3, 3) símplices que, por lo que sabemos, era desconocida hasta ahora.

El lector puede comprender que este conjunto de cálculos requiere una compleja interacción por parte del usuario, en particular invocando explícitamente \mathbb{Z} -ciclos, y dicha funcionalidad no puede ser incorporada de un modo sencillo en el marco diseñado para *fKenzo*.

Se puede encontrar otro ejemplo de este mismo estilo en el artículo [1]. Una A_∞ -estructura de un complejo de cadenas C_* es una multiplicación μ_2 definida sobre C_* , compatible con la diferencial, pero asociativa sólo salvo homotopía. Entonces se requiere una homotopía explícita $\mu_3 : C_* \otimes C_* \otimes C_* \rightarrow C_*$, que a su vez debe verificar la propiedad asociativa salvo homotopía *explícita* μ_4 , etcétera. Para una descripción precisa de esta estructura relativamente compleja, descubierta por Jim Stasheff en los años 60 (ver [11]), recomendamos la referencia [6].

Una A_∞ -estructura es compleja, pero puede ser fácilmente construida en algunos contextos por el programa *Kenzo*, gracias a la potencia del Lema de Perturbación Básico. El artículo [1] explica cómo ciertas A_∞ -estructuras no triviales se pueden construir en *Kenzo* cuando se aplican, de nuevo, métodos de la homología efectiva para el cálculo de $H_*(\Omega^3 P^\infty \mathbb{R}/P^3 \mathbb{R})$. A algunos complejos de cadenas intermedios que son necesarios para el cálculo de estos grupos de homología se les puede dotar de una A_∞ -estructura $(\mu_n)_{n \geq 1}$, y un análisis cuidadoso de dicha estructura muestra que μ_n es no trivial. Esto requiere un estudio de los términos que componen $\mu_n(g \otimes \cdots \otimes g)$, y no se puede proporcionar un análisis tan meticuloso a través de una interfaz gráfica, al menos de un modo fácil y *usable* (es decir, sin dar acceso a las estructuras internas de *Common Lisp*).

5. CONCLUSIONES Y TRABAJO FUTURO

fKenzo es una interfaz de usuario para el sistema *Kenzo*, un programa *Common Lisp* que permite realizar cálculos en Topología Algebraica. En su estado actual, *fKenzo* satisface dos objetivos: por un lado proporciona una interfaz amigable para *Kenzo*; y, por otro, guía al usuario para evitar errores de ejecución, provenientes tanto de decisiones de diseño de *Kenzo* como de cuestiones topológicas. Este segundo objetivo se consigue gracias a un programa *mediador* llamado *capa intermedia*. Esperamos que estos sean unos pasos suficientemente firmes hacia nuestro objetivo final, que no es otro que incrementar el interés de topólogos algebraicos hacia *Kenzo*, o, más en general, hacia las aproximaciones constructivas y efectivas a la Topología Algebraica.

Tres grandes líneas de futuro permanecen abiertas.

- La primera está relacionada con incluir más funcionalidades de *Kenzo* en *fKenzo*. Uno de los mayores retos consiste en encontrar un modo adecuado (libre de la sintaxis *Common Lisp*) de editar y manejar los elementos de los espacios construidos. De este modo podríamos aproximarnos a la difícil cuestión de introducir en *fKenzo* cálculos como los presentados en la sección 4.
- La segunda línea de trabajo está relacionada con la extensibilidad de *fKenzo*. Dado que su núcleo computacional, *Kenzo*, continúa creciendo cubriendo cada vez más aspectos de la Topología Algebraica (ver por ejemplo el trabajo [8] donde se presenta una extensión de *Kenzo* para sucesiones espectrales), es necesario que *fKenzo* pueda evolucionar en consecuencia. La estructura modular de *fKenzo*, basada en mecanismos *OpenMath*, será fundamental en esta tarea.
- Finalmente, nos gustaría que *fKenzo* se convirtiera en un asistente integral para la Topología Algebraica, incluyendo cálculo (el único aspecto considerado hasta ahora), comunicación (con otros sistemas) y deducción (por medio de asistentes para la demostración). Ya comentamos previamente en este artículo que la distribución actual de *fKenzo* dispone de unos desarrollos preliminares para conectarlo con el sistema de cálculo simbólico *GAP* y el demostrador de teoremas *ACL2*.

REFERENCIAS

- [1] A. BERCIANO, J. RUBIO Y F. SERGERAERT, A case study of A_∞ -structure, *Georgian Math. J.* **17** (2010), 57–77.
- [2] S. BUSWELL, O. CAPROTTI, D. P. CARLISLE, M. C. DEWAR, M. GAÉTANO Y M. KOHLHASE, OpenMath, versión 2.0, 2004. Disponible en <http://www.openmath.org/>.
- [3] X. DOUSSON, F. SERGERAERT Y Y. SIRET, The Kenzo program, 1999. Disponible en <http://www-fourier.ujf-grenoble.fr/~sergerar/Kenzo/>.
- [4] J. HERAS, *fKenzo*, 2010. Disponible en <http://www.unirioja.es/cu/joheras/>.
- [5] J. HERAS, V. PASCUAL Y J. RUBIO, Mediated Access to Symbolic Computation Systems, *Lecture Notes in Artificial Intelligence* **5144** (2008), 446–461.
- [6] T. KADEISHVILI, Operadic Algebraic Topology, *Ictp-Map 2008 Summer School*, 2008. Disponible en http://map.disi.unige.it/ictp/lectures_files/Kadeishvili_L.pdf.
- [7] J. P. MAY, *Simplicial Objects in Algebraic Topology*, Van Nostrand, 1967.
- [8] A. ROMERO, J. RUBIO Y F. SERGERAERT, Computing Spectral Sequences, *J. Symbolic Comput.* **41** (2006), 10:1059–1079.
- [9] J. RUBIO Y F. SERGERAERT, Constructive algebraic topology, *Bull. Sci. Math.* **126** (2002), 5:389–412.
- [10] F. SERGERAERT, Triangulations of complex projective spaces, en *Contribuciones científicas en honor de Mirian Andrés Gómez*, Universidad de La Rioja, 2010, 507–519. Disponible en <http://www.unirioja.es/servicios/sp/catalogo/monografias/vr77.shtml>.
- [11] J. D. STASHEFF, Homotopy associativity of H -spaces, I, II, *Trans. Amer. Math. Soc.* **108** (1963), 275–292 y 293–312.

J. HERAS, DPTO. DE MATEMÁTICAS Y COMPUTACIÓN, UNIVERSIDAD DE LA RIOJA, LOGROÑO
 Correo electrónico: jonathan.heras@unirioja.es
 Página web: <https://esus.unirioja.es/psycotrip/>

V. PASCUAL, DPTO. DE MATEMÁTICAS Y COMPUTACIÓN, UNIVERSIDAD DE LA RIOJA, LOGROÑO
 Correo electrónico: vico.pascual@unirioja.es
 Página web: <https://esus.unirioja.es/psycotrip/>

J. RUBIO, DPTO. DE MATEMÁTICAS Y COMPUTACIÓN, UNIVERSIDAD DE LA RIOJA, LOGROÑO
 Correo electrónico: julio.rubio@unirioja.es
 Página web: <https://esus.unirioja.es/psycotrip/>

F. SERGERAERT, INSTITUT FOURIER, UNIVERSITÉ JOSEPH FOURIER, GRENOBLE, FRANCIA
 Correo electrónico: francis.sergeraert@ujf-grenoble.fr
 Página web: <http://www-fourier.ujf-grenoble.fr/~sergerar/>