

L^AT_EX Source for GR6

Contents

1	Introduction	1	3.4	Delimiters	7
1.1	Previous version (GR5)	1	3.5	Operators	7
1.2	Present version	2	3.6	Miscellaneous things	7
2	Discussion of code	2	4	LaTeX source: high-level code for formatting	8
3	LaTeX source: low-level code for expressions	2	4.1	How information was entered	8
3.1	Macro naming convention for function representation	3	4.2	Dimensions	9
3.1.1	Detailed functions and their arguments	3	4.3	Other	9
3.1.2	Functions and their arguments	3	5	Sample Source Code	10
3.1.3	Special functions	4	5.1	Source Code	10
3.1.4	Derivative and integral functions	4	5.2	Formatted output	11
3.1.5	Sum and product functions	4	6	System level details of GR6	12
3.2	Constants	4	6.1	L ^A T _E X packages	12
3.3	Function names	5	6.2	Directories and files	12
			6.2.1	GR/STYLES files	12
			6.2.2	GR/FINAL files	13
			6.2.2.1	GR6 source	13
			6.2.2.2	The shell scripts	13
			6.2.2.3	Other files	14

1 Introduction

This document describes the L^AT_EX source for the sixth edition of Gradshteyn and Ryzhik (denoted GR6).

1.1 Previous version (GR5)

The electronic source of the fifth edition of GR (denoted GR5) was made available from Technical Typesetters. This source consisted of inelegant T_EX code (not L^AT_EX) along with a lengthy collection of macros. Difficulties with this source included:

1. The marked up T_EX was not as clear as it could be. For example, the same command means different things in different files:
 - `jeffc00.tex` has `\def\th{\mathop{\rm th}\nolimits}`
 - `jeffc07b.tex` has `\def\th{\theta}`
2. All of the integrals were displayed using T_EX's primitive `\displaylines` command.
3. The mathematics, as written in T_EX, was frequently ambiguous.
4. There were many implicit multiplications in the expressions.
5. There were numerous "obvious" typographical errors.
6. The fonts used were proprietary.
7. There was no understanding in the files of the mathematical semantics of what was being typeset. For example, expression 2.01.13 is represented, in part, by line 243 of `jeffc02a.tex` which has:
`\ln {\rm t}{\rm g}{x\over 2}`.

That is, this is the logarithm of the tangent (the notation "tg" was used for tangent) of x over 2. Unfortunately, the tangent is indicated by two letters which have no clear relation to any function and the argument of neither the logarithm nor the tangent are clearly indicated.

(Just for comparison, in GR6 this expression is represented as $\backslash\text{ZFUNCTIONArg}\{\backslash\text{ZFUNCTIONln}\}\{\backslash\text{ZFUNCTIONArg}\{\backslash\text{ZFUNCTIONtg}\}\{\backslash\text{Zfrac}\{x\}\{2\}\}\}$. Here the argument of each mathematical function is clearly indicated.)

8. The special functions are not distinguished in the source. For example, B-with-a-subscript is used for all of a Bernoulli number, a Bernoulli polynomial, and an incomplete Beta function. (This makes it hard to electronically archive the results based on the types of functions appearing in each integral.)
9. They had a Russian style line breaking: “=”, “+”, “−”, and “×” appeared at the end of a line to be broken and also at the beginning of the next line.
10. They used a non-standard mathematical notation: “th” for “tanh”, “sh” for “sinh”, “ch” for “cosh”, etc
11. Single chapters were spread across multiple files.

1.2 Present version

We created an object model for the expressions in GR6, and then forced all input to the object model. Lots of “cleaning up” was also performed. For example: all occurrences of `\over` were replaced with a fraction command taking two arguments (such as `\frac`).

2 Discussion of code

The goal was to replace the $\text{T}_{\text{E}}\text{X}$ code with $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ code that could be easily formatted for the printed page, but would also contain all of the semantic information for each expression. The new “master electronic source”:

- Could render the mathematics into print format, similar to it’s present appearance.
- Would be mathematically unambiguous.
- Would be easy to translate the existing source into.

As a result, there are two levels of macros. The low-level is for the representation of each sub-expression; the high-level code is for nicely formatting the information in GR6.

The collections of macro are in the files:

- `gr.macros.all`: main one (calls all the others)
- `gr.macros.diagnostic`: (optional)
- `gr.macros.math.operators`
- `gr.macros.object_model`
- `gr.macros.other`

3 LaTeX source: low-level code for expressions

Note that many macros names begin with the letter “Z”.

3.1 Macro naming convention for function representation

Numerous commands were defined in L^AT_EX to represent expressions. These expressions were semantically complete and can be easily translated into symbolic manipulation languages.

The macros names are mostly created from a set of key words strung together. These keywords include:

- `ArbLet` for arbitrary letters
- `Arg` for arguments
- `Power` for powers
- `Prime` for primes (')
- `Prod` for products
- `Sub` for subscripts
- `Sum` for sums
- `Sup` for superscript
- `ZFUNCTION ...` for introducing a function command or a function

The order in which these keywords appears indicates the order of the arguments to the macro. For example, `\ZFUNCTIONSupSubArg` takes four arguments, a function, a subscript, a superscript, and an argument. As an illustration* `\ZFUNCTIONSupSubArg{\ZFUNCTIONHankel}{2}{3}{(\alpha x)}` would appear as $H_3^2(\alpha x)$.

3.1.1 Detailed functions and their arguments

These constructs include:

- `\ZPower<argument> <power>`
For example: `\Zpower{(\alpha x)}{2}` would appear as $(\alpha x)^2$
- `\ZFUNCTIONArg<name of function> <argument to function>`
For example: `\ZFUNCTIONArg{\ZFUNCTIONsin}{\alpha x}` would appear as $\sin \alpha x$
- `\ZFUNCTIONPowerArg<name of function> <argument to function> <power to function>`
For example: `\ZFUNCTIONPowerArg{\ZFUNCTIONsin}{2}{\alpha x}` would appear as $\sin^2 \alpha x$
- `\ZFUNCTIONSubArg<name of function> <subscript> <argument to function>`
For example: `\ZFUNCTIONSubArg{\ZFUNCTIONBesselJ}{2}{\Zsurround{\alpha x}}` would appear as $J_2(\alpha x)$
- `\ZFUNCTIONSupSubArg<name of function> <superscript> <subscript> <argument to function>`
For example: `\ZFUNCTIONSupSubArg{\ZFUNCTIONHankel}{2}{3}{\Zsurround{\alpha x}}` would appear as $H_3^2(\alpha x)$

3.1.2 Functions and their arguments

- `\ZArbLetPrimeSub{A}{B}{C}` would appear as A'_B
- `\ZArbLetSubSup{A}{B}{C}` would appear as A_B^C
- `\ZFUNCTIONArg{A}{(B)}` would appear as $A(B)$
- `\ZFUNCTIONPowerArg{A}{B}{(C)}` would appear as $A^B(C)$
- `\ZFUNCTIONPowerSubArg{A}{B}{C}{(D)}` would appear as $A_C^B(D)$
- `\ZFUNCTIONPowerSubArg{A}{B}{C}{(D)}` would appear as $A_C^B(D)$
- `\ZFUNCTIONPrimeArg{A}{B}` would appear as $A'B$

*All the terms in this expression are defined in subsequent sections.

- `\ZFUNCTIONPrimePrimeArg{A}{B}` would appear as $A''B$
- `\ZFUNCTIONPrimePrimePrimeArg{A}{B}` would appear as $A'''B$
- `\ZFUNCTIONPrimePrimePrimeSubArg{A}{B}{C}` would appear as $A_B'''(C)$
- `\ZFUNCTIONPrimePrimeSubArg{A}{B}{C}` would appear as $A_B''(C)$
- `\ZFUNCTIONPrimeSubArg{A}{B}{C}` would appear as $A_B'(C)$
- `\ZFUNCTIONSubArg{A}{B}{C}` would appear as $A_B(C)$
- `\ZFUNCTIONSubPowerArg{A}{B}{C}{D}` would appear as $A_B^C(D)$
- `\ZFUNCTIONSubPowerArg{A}{B}{C}{D}` would appear as $A_B^C(D)$
- `\ZFUNCTIONSubSupArg{A}{B}{C}{D}` would appear as $A_B^C(D)$
- `\ZFUNCTIONSubSupArg{A}{B}{C}{D}` would appear as $A_B^C(D)$
- `\ZFUNCTIONSupSubArg{A}{B}{C}{D}` would appear as $A_C^B(D)$
- `\ZFUNCTIONSupSubArg{A}{B}{C}{D}` would appear as $A_C^B(D)$

3.1.3 Special functions

The hypergeometric function and the Meijer G function were handled separately:

- For example: `\ZFUNCTIONMeijerGWithArgs{A}{B}{C}{D}{E}` would appear as $G_A^B \left(C \middle| \begin{matrix} D \\ E \end{matrix} \right)$
- For example: `\ZFUNCTIONHypergeometricFunctionWithArgs{A}{B}{C}{D}{E}` would appear as ${}_A F_B (C; D; E)$

3.1.4 Derivative and integral functions

- `\ZIntegralIndefinite` would appear as \int
- `\ZIntegralLowerUpper{A}{B}` would appear as \int_A^B
- `\diff{A}{B}` would appear as $\frac{dA}{dB}$
- `\mdiff{A}{B}{C}` would appear as $\frac{d^C A}{dB^C}$
- `\mpdiff{A}{B}{C}` would appear as $\frac{\partial^C A}{\partial B^C}$
- `\pdiff{A}{B}` would appear as $\frac{\partial A}{\partial B}$

3.1.5 Sum and product functions

- `\ZProdPrimeSubSup{A}{B}` would appear as $\prod_A'^B$
- `\ZProdSubSup{A}{B}` would appear as \prod_A^B
- `\ZProdSub{A}` would appear as \prod_A
- `\ZProdSupSub{A}{B}` would appear as \prod_B^A
- `\ZSumPrimeSubSup{A}{B}` would appear as $\sum_A'^B$
- `\ZSumSubSup{A}{B}` would appear as \sum_A^B
- `\ZSumSub{A}` would appear as \sum_A
- `\ZSumprime` would appear as \sum'

3.2 Constants

The following constants were defined:

1.	<i>G</i>	<code>\ConstantCatalan</code>	6.	<i>e</i>	<code>\ConstantE</code>
2.	<i>D</i>	<code>\ConstantEllipticFunctionDz</code>	7.	∞	<code>\ConstantInfy</code>
3.	<i>E</i>	<code>\ConstantEllipticFunctionEz</code>	8.	<i>i</i>	<code>\Constanti</code>
4.	<i>K</i>	<code>\ConstantEllipticFunctionKz</code>	9.	π	<code>\Constantpi</code>
5.	<i>C</i>	<code>\ConstantEuler</code>			

The following constants were indexed, each took one argument (which appeared as a subscript):

- `\CONSTANTBernoulli{n}` would appear as B_n
- `\CONSTANTBernoulliMod{n}` would appear as B_n^*
- `\CONSTANTEuler{n}` would appear as E_n

3.3 Function names

The function names were changed from being single letters (or short names), which were sometime ambiguous, into long description names.

1.	Ai	<code>\MATHAi</code>	36.	Fe	<code>\MATHFe</code>
2.	J	<code>\MATHAngerFunction</code>	37.	<i>C</i>	<code>\MATHFresnelCosineIntegral</code>
3.	arcosech	<code>\MATHArcosech</code>	38.	<i>S</i>	<code>\MATHFresnelSineIntegral</code>
4.	arcsech	<code>\MATHArsech</code>	39.	Γ	<code>\MATHGammaIncomplete</code>
5.	<i>P</i>	<code>\MATHAssociatedLegendreFunctionFirst</code>	40.	Γ	<code>\MATHGamma</code>
6.	<i>Q</i>	<code>\MATHAssociatedLegendreFunctionSecond</code>	41.	<i>C</i>	<code>\MATHGegenbauerPolynomial</code>
7.	<i>k</i>	<code>\MATHBatemanFunction</code>	42.	Gek	<code>\MATHGek</code>
8.	<i>B</i>	<code>\MATHBernoulliPolynomial</code>	43.	Gey	<code>\MATHGey</code>
9.	<i>I</i>	<code>\MATHBesselI</code>	44.	Ge	<code>\MATHGe</code>
10.	<i>J</i>	<code>\MATHBesselJ</code>	45.	<i>H</i>	<code>\MATHHTheta</code>
11.	<i>K</i>	<code>\MATHBesselK</code>	46.	<i>H</i>	<code>\MATHHankel</code>
12.	<i>Y</i>	<code>\MATHBesselY</code>	47.	<i>H</i>	<code>\MATHHeavisideFunction</code>
13.	\mathfrak{Z}	<code>\MATHBesselZFraktur</code>	48.	<i>He</i>	<code>\MATHHermitePolynomialHe</code>
14.	<i>Z</i>	<code>\MATHBesselZ</code>	49.	<i>H</i>	<code>\MATHHermitePolynomial</code>
15.	<i>Z</i>	<code>\MATHBessel</code>	50.	chi	<code>\MATHHyperbolicCosineIntegral</code>
16.	<i>B</i>	<code>\MATHBeta</code>	51.	shi	<code>\MATHHyperbolicSineIntegral</code>
17.	<i>O</i>	<code>\MATHBigO</code>	52.	Im	<code>\MATHIm</code>
18.	Bi	<code>\MATHBi</code>	53.	B	<code>\MATHIncompleteBetaB</code>
19.	Ce	<code>\MATHCe</code>	54.	<i>I</i>	<code>\MATHIncompleteBetaI</code>
20.	<i>T</i>	<code>\MATHChebyshevPolynomialOne</code>	55.	<i>P</i>	<code>\MATHJacobiPolynomial</code>
21.	<i>U</i>	<code>\MATHChebyshevPolynomialTwo</code>	56.	<i>L</i>	<code>\MATHLaguerrePolynomial</code>
22.	Cin	<code>\MATHCin</code>	57.	<i>L</i>	<code>\MATHLaguerre</code>
23.	Ci	<code>\MATHCi</code>	58.	<i>P</i>	<code>\MATHLegendreFunctionFirst</code>
24.	Ei	<code>\MATHEi</code>	59.	<i>Q</i>	<code>\MATHLegendreFunctionSecond</code>
25.	<i>D</i>	<code>\MATHEllipticFunctionD</code>	60.	Φ	<code>\MATHLerch</code>
26.	E	<code>\MATHEllipticFunctionEComplete</code>	61.	<i>L</i>	<code>\MATHLobachevskiyFunction</code>
27.	<i>E</i>	<code>\MATHEllipticFunctionE</code>	62.	<i>S</i>	<code>\MATHLommelFunctionS</code>
28.	<i>F</i>	<code>\MATHEllipticFunctionF</code>	63.	<i>U</i>	<code>\MATHLommelFunctionU</code>
29.	K	<code>\MATHEllipticFunctionKComplete</code>	64.	<i>V</i>	<code>\MATHLommelFunctionV</code>
30.	<i>K</i>	<code>\MATHEllipticFunctionK</code>	65.	<i>s</i>	<code>\MATHLommelFunctions</code>
31.	Π	<code>\MATHEllipticFunctionPi</code>	66.	<i>E</i>	<code>\MATHMacRobertE</code>
32.	<i>E</i>	<code>\MATHEulerPolynomial</code>	67.	L	<code>\MATHModifiedStruve</code>
33.	<i>E</i>	<code>\MATHExponentialIntegral</code>	68.	<i>O</i>	<code>\MATHNeumannPolynomial</code>
34.	Fek	<code>\MATHFek</code>	69.	<i>D</i>	<code>\MATHParabolicCylinder</code>
35.	Fey	<code>\MATHFey</code>	70.	Φ	<code>\MATHPhiConfluentHypergeometric</code>

71.	Φ	<code>\MATHPhi</code>	115.	ds	<code>\MATHds</code>
72.	Ψ	<code>\MATHPsiConfluentHypergeometric</code>	116.	erfc	<code>\MATHerfc</code>
73.	Re	<code>\MATHRe</code>	117.	erf	<code>\MATHerf</code>
74.	S	<code>\MATHSchlafliPolynomial</code>	118.	exp	<code>\MATHexp</code>
75.	Se	<code>\MATHSe</code>	119.	fe	<code>\MATHfe</code>
76.	Si	<code>\MATHSi</code>	120.	γ	<code>\MATHgammaIncomplete</code>
77.	H	<code>\MATHStruve</code>	121.	γ	<code>\MATHgamma</code>
78.	Θ	<code>\MATHTheta</code>	122.	gd	<code>\MATHgd</code>
79.	E	<code>\MATHWeberFunction</code>	123.	ge	<code>\MATHge</code>
80.	M	<code>\MATHWhittakerFunctionM</code>	124.	hei	<code>\MATHhei</code>
81.	W	<code>\MATHWhittakerFunctionW</code>	125.	her	<code>\MATHher</code>
82.	C	<code>\MATHYoung</code>	126.	kei	<code>\MATHkei</code>
83.	am	<code>\MATHam</code>	127.	ker	<code>\MATHker</code>
84.	arccosec	<code>\MATHarccosec</code>	128.	λ	<code>\MATHlambda</code>
85.	arccosh	<code>\MATHarccosh</code>	129.	li	<code>\MATHli</code>
86.	arccos	<code>\MATHarccos</code>	130.	ln	<code>\MATHln</code>
87.	arccoth	<code>\MATHarccoth</code>	131.	log	<code>\MATHlogBase</code>
88.	arccot	<code>\MATHarccot</code>	132.	log	<code>\MATHlog</code>
89.		<code>\MATHarcctg</code>	133.	μ	<code>\MATHmu</code>
90.	arcsec	<code>\MATHarcsec</code>	134.	nc	<code>\MATHnc</code>
91.	arcsinh	<code>\MATHarcsinh</code>	135.	nd	<code>\MATHnd</code>
92.	arcsin	<code>\MATHarcsin</code>	136.	ns	<code>\MATHns</code>
93.	arctanh	<code>\MATHarctanh</code>	137.	ν	<code>\MATHnu</code>
94.	arctan	<code>\MATHarctan</code>	138.	ψ	<code>\MATHpsi</code>
95.	arg	<code>\MATHarg</code>	139.	sc	<code>\MATHsc</code>
96.	bei	<code>\MATHbei</code>	140.	sd	<code>\MATHsd</code>
97.	ber	<code>\MATHber</code>	141.	sech	<code>\MATHsech</code>
98.	β	<code>\MATHbetaFunction</code>	142.	sec	<code>\MATHsec</code>
99.	β	<code>\MATHbeta</code>	143.	se	<code>\MATHse</code>
100.	cd	<code>\MATHcd</code>	144.	sgn	<code>\MATHsgn</code>
101.	ce	<code>\MATHce</code>	145.	σ	<code>\MATHsigma</code>
102.	chi	<code>\MATHchi</code>	146.	sign	<code>\MATHsign</code>
103.	ci	<code>\MATHci</code>	147.	sinh	<code>\MATHsinh</code>
104.	cn	<code>\MATHcn</code>	148.	sin	<code>\MATHsin</code>
105.	cosech	<code>\MATHcosech</code>	149.	si	<code>\MATHsi</code>
106.	cosec	<code>\MATHcosec</code>	150.	sn	<code>\MATHsn</code>
107.	cosh	<code>\MATHcosh</code>	151.	tanh	<code>\MATHtanh</code>
108.	cos	<code>\MATHcos</code>	152.	tan	<code>\MATHtan</code>
109.	coth	<code>\MATHcoth</code>	153.	ϑ	<code>\MATHtheta</code>
110.	cot	<code>\MATHcot</code>	154.	tr	<code>\MATHtrace</code>
111.	cs	<code>\MATHcs</code>	155.	\wp	<code>\MATHwp</code>
112.	dc	<code>\MATHdc</code>	156.	ξ	<code>\MATHxi</code>
113.	δ	<code>\MATHdelta</code>	157.	ζ	<code>\MATHzeta</code>
114.	dn	<code>\MATHdn</code>	158.	zn	<code>\MATHzn</code>

Corresponding to each function name of the form `\Mathfoo` there was also a function `\ZFUNCTIONfoo`. This second form was used in GR6. It called the `\Mathfoo` macro for the actual representation of the function, sent information to the index, etc.

3.4 Delimiters

Parenthesis, brackets, and the like that are applied to expressions were replaced with commands (many grow in size as needed):

1.	(a)	<code>\Zsurround{a}</code>	6.	a	<code>\Znorm{a}</code>
2.	$[a]$	<code>\Zsurbrack{a}</code>	7.	$ a $	<code>\Zsurvert{a}</code>
3.	$\{a\}$	<code>\Zsurbrace{a}</code>	8.	$a!$	<code>\ZFactorial{a}</code>
4.	$ a $	<code>\Zabs{a}</code>	9.	$a!!$	<code>\ZFactorialDouble{a}</code>
5.	$ a $	<code>\Zdeterminant{a}</code>			

3.5 Operators

The following operators are defined (the textual representation is followed by the macro name):

1.	H	<code>\Hermitian</code>	6.	curl	<code>\curl</code>
2.	PV	<code>\PrincipalValue</code>	7.	div	<code>\div</code>
3.	T	<code>\Transpose</code>	8.	grad	<code>\grad</code>
4.	adj	<code>\Zadj</code>	9.	rot	<code>\rot</code>
5.	∇	<code>\Znabla</code>	10.	tr	<code>\tr</code>

3.6 Miscellaneous things

For ease of typing, letters that were bolded, or had bars over them were represented by simple macros:

1.	$\bar{\xi}$	<code>\barxi</code>	20.	b	<code>\bfb</code>	39.	y	<code>\bfy</code>
2.	\bar{z}	<code>\barz</code>	21.	c	<code>\bfc</code>	40.	0	<code>\bfzero</code>
3.	A	<code>\bfA</code>	22.	e	<code>\bfe</code>	41.	z	<code>\bfz</code>
4.	B	<code>\bfB</code>	23.	f	<code>\bff</code>	42.	F	<code>\capF</code>
5.	C	<code>\bfC</code>	24.	g	<code>\bfg</code>	43.	I	<code>\capI</code>
6.	H	<code>\bfH</code>	25.	i	<code>\bfi</code>	44.	\hat{x}	<code>\hatx</code>
7.	I	<code>\bfI</code>	26.	j	<code>\bfj</code>	45.	F_c	<code>\FourierCosineFunction</code>
8.	K	<code>\bfK</code>	27.	k	<code>\bfk</code>	46.	\mathcal{F}_c	<code>\FourierCosineTransform</code>
9.	M	<code>\bfM</code>	28.	λ	<code>\bflambda</code>	47.	F_s	<code>\FourierSineFunction</code>
10.	P	<code>\bfP</code>	29.	μ	<code>\bfmu</code>	48.	\mathcal{F}_s	<code>\FourierSineTransform</code>
11.	Q	<code>\bfQ</code>	30.	m	<code>\bfm</code>	49.	\mathcal{F}	<code>\FourierTransform</code>
12.	R	<code>\bfR</code>	31.	n	<code>\bfn</code>	50.	\mathfrak{S}	<code>\FrakturS</code>
13.	S	<code>\bfS</code>	32.	ϕ	<code>\bfphi</code>	51.	\mathfrak{Z}	<code>\FrakturZ</code>
14.	U	<code>\bfU</code>	33.	q	<code>\bfq</code>	52.	\mathcal{L}	<code>\LaplaceTransform</code>
15.	V	<code>\bfV</code>	34.	r	<code>\bfr</code>	53.	\mathcal{M}	<code>\MellinTransform</code>
16.	W	<code>\bfW</code>	35.	u	<code>\bfu</code>	54.	\mathbb{N}	<code>\ZNaturalNumbers</code>
17.	X	<code>\bfX</code>	36.	v	<code>\bfv</code>	55.	\mathbb{Z}	<code>\Zinteger</code>
18.	Y	<code>\bfY</code>	37.	w	<code>\bfw</code>	56.	\mathbb{N}	<code>\Znatural</code>
19.	a	<code>\bfa</code>	38.	x	<code>\bfx</code>	57.	\mathbb{R}	<code>\Zreal</code>

Other things:

1.	κ	<code>\curlykappa</code>	4.		<code>\pminus</code>
2.	k'	<code>\kPrime</code>	5.	κ	<code>\varkappa</code>
3.	n'	<code>\nPrime</code>	6.	\circ	<code>\zdegrees</code>

4 LaTeX source: high-level code for formatting

The \LaTeX macros used to create GR6 recognize that an object has one or more of the following components:

- a number (stored away with `\ZDefNumber{...}`)
- an integral (stored away with `\ZDefIntegral{...}`)
- a constraint (stored away with `\ZDefConstraint{...}`)
- a reference (stored away with `\ZDefReference{...}`)
- a left hand side to an expression (stored away with `\ZDefLHS{...}`)
- a right hand side to an expression (stored away with `\ZDefRHS{...}`)

If an integral had 4 right hand sides, then they would be stored away with the commands `\ZDefrhsONE{...}`, `\ZDefrhsTWO{...}`, `\ZDefrhsTHREE{...}`, and `\ZDefrhsFOUR{...}`. In general, for those things that there could one of more or, we sometimes used a suffix of ONE, TWO, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT, NINE, X, XI, XII, for the “first occurrence of something” through to the “twelfth occurrence of something”.

4.1 How information was entered

Typically, for integrals in a list, an entire expression might look like:

```
\Zitem
\ZDefIntegral{...}
\ZDefReference{...}
\ZDefConstraint{...}
\ZZdoit
```

where not all the components need be present, and the order (aside from the `\Zitem` and `\ZZdoit`) is irrelevant. Another form might be

```
\Zitem
\ZDefLHS{...}
\ZDefrhsONE{= ...}
\ZDefConstraintONE{...}
\ZDefrhsTWO{=...}
\ZDefrhsTHREE{\ZContinuationSpacing + ...} = ...}
\ZDefrhsFOUR{\ZContinuationSpacing -...}
\ZDefReferenceFOUR{...}
\ZDefConstraintFOUR{...}
\ZZdoit
```

In this case, there are two right hand sides, one takes a single line and one takes three lines.

The macro `\ZZdoit` chooses the most appropriate formatting for the given expression. This is done, in part, by defining many dimensions (see next section).

4.2 Dimensions

The control of the output was controlled by many lengths that were defined in the program. These included:

- | | | | | | |
|----|------------|---------------------------------------|-----|-------------|--|
| 1. | 10.84006pt | <code>\WidthCONtoRightMargin</code> | 9. | 7.22743pt | <code>\WidthListToINT</code> |
| 2. | 10.84006pt | <code>\WidthCONtoREF</code> | 10. | 7.22743pt | <code>\WidthMultiLineToRightMargin</code> |
| 3. | 36.135pt | <code>\WidthLeftMarginToText</code> | 11. | 420.06937pt | <code>\WidthOfTextMinusLeftMarginToText</code> |
| 4. | 14.45377pt | <code>\WidthINTtoCON</code> | 12. | 57.81621pt | <code>\WidthCON</code> |
| 5. | 14.45377pt | <code>\WidthINTtoREF</code> | 13. | 108.405pt | <code>\WidthREF</code> |
| 6. | 10.84006pt | <code>\WidthRHStoREF</code> | 14. | 177.06126pt | <code>\WidthCONplusREF</code> |
| 7. | 72.26999pt | <code>\WidthItemBoxInZitemList</code> | 15. | 01em | <code>\ZContinuationSpacing</code> |
| 8. | 10.84006pt | <code>\WidthLeftofNumberInList</code> | | | |

For example, the dimension `WidthCONtoREF` was the space between a constraint and a reference when they appeared on the same line.

There were also temporary length registers used for computational purposes:

1. `GlobalTempTextWidthONE`,
2. `GlobalTempTextWidthTWO`,
3. `WidthTemp`, and
4. `WidthTempTwo`.
5. `WidthToFitRHSIn`
6. `NumWidthToFitIntConRef`

And there were dimensions that were determined directly from the input expressions:

1. `WidthLHS`
2. `WidthMultiRHSS`
3. `WidthMultiConstraints`
4. `WidthMultiReferences`

4.3 Other

1. There are flags that can be turned on to follow how `\ZZdoit` decides how to format the information. Essentially computations were performed to determine how the data should appear (i.e., all on one line or on more than one line) and then a macro for exactly that type of formatting was called. Some of the macro names include (the word “Local” is used for macros that are not called by the source, but only by other macros):
 - (a) `LocalZFormatPossibleGlobalConAndRef`
 - (b) `LocalZListIntConaRef`
 - (c) `LocalZListIntCon`
 - (d) `LocalZListIntRef`
 - (e) `LocalZListInt`
 - (f) `LocalZListMultilineZwithRHSandCONandREF`
 - (g) `LocalZListMultilineZwithRHSandCONnoREF`
 - (h) `LocalZListMultilineZwithRHSnoCONandREF`
 - (i) `LocalZListMultilineZwithRHSnoCONnoREF`
 - (j) `LocalZListSingle`
2. Of course, all the sectioning routines were re-defined so that the results would set up the appropriate running heads and the like. The list of such commands is

(a) <code>\ZchapterSTAR</code>	(g) <code>\ZsubsectionSHORTtoc</code>
(b) <code>\Zchapter</code>	(h) <code>\ZsubsectionSHORT</code>
(c) <code>\ZsectionSHORT</code>	(i) <code>\ZsubsectionWithStar</code>
(d) <code>\ZsectionWithStar</code>	(j) <code>\Zsubsection</code>
(e) <code>\Zsectiontoc</code>	(k) <code>\Zsubsubsection</code>
(f) <code>\Zsection</code>	

- For expressions that were not within a list, (so that `\Zitem` was not used) the commands following the definitions of all the terms was either `\ZZdoit` or `\ZZCenter` (in which case the output should be centered).
- Within the macro source, the following abbreviations were used:
 - Num for “Number”
 - Con for “Constraint”
 - Ref for “Reference”
 - Int for “Integral”
- A new section used the number defined via `\ZDefNumber`. If text followed the section number than `\ZZNumberThenText` was used. If a list followed the section number, than `\ZZNumberThenList` was used.
- Sometimes a small superscript indicating a revision had to be included. This was defined by using of `\ModifiedTextSet` before `\ZZitem` or `\ZZNumberThenText` or `\ZZNumberThenList` For example, using `\ModifiedTextSet{8}` would result in a superscript of “8” appearing on an integral number, or on a section number.
- There are a lot more details!

5 Sample Source Code

5.1 Source Code

The following sample source code:

```

1 \ZDefNumber{3.136}
2 \ZZNumberThenList
3 \ZbeginE
4
5 \Zitem
6 %
7 \ZDefLHS{
8 {\ZIntegralLowerUpper{-\Constantinfty}{u}}
9 \Zfrac{\Zdifferential{dx}}{\ZFUNCTIONSqrt{\Zpower{(a-x)}{3}}
10 \Zpower{(b-x)}{3}}
11 \Zpower{(c-x)}{3}}}}
12 \ZDefrhsONE{
13     =\Zfrac{2}{\Zpower{(a-b)}{2}}
14 \Zpower{(b-c)}{2} \ZFUNCTIONSqrt{\Zpower{(a-c)}{3}}}}
15 }
16 %
17 \ZDefrhsTWO{
18 \ZContinuationSpacing
19 \times
20 \Zsurbrack{(b-c)(a+b-2c)}
21 \ZFUNCTIONArg{\ZFUNCTIONEllipticFunctionFz}{(\alpha, p)}-2
22 \Zsurround{\Zpower{c}{2}+\Zpower{a}{2}+\Zpower{b}{2}-ab-ac-bc}
23 \ZFUNCTIONArg{\ZFUNCTIONEllipticFunctionEz}{(\alpha, p)}}}

```

```

24 %
25 \ZDefrhsTHREE{
26 \ZContinuationSpacing
27 +\Zfrac{2[c(a-c)+b(a-b)-u(2a-c-b)]}{(a-b)(a-c)}
28 \Zpower{(b-c)}{2} \ZFUNCTIONSqrt{(a-u)(b-u)(c-u)}}
29 }
30 %
31 \ZDefConstraint{ \Zsurbrack{a>b>c>u}}
32 %
33 \ZDefReference{BY (231.14)}
34 %
35 \ZZdoit
36
37 \Zitem
38 \ZDefLHS{
39 {\ZIntegralLowerUpper{u}{\Constantinfty}}
40 \Zfrac{\Zdifferential{dx}}{\ZFUNCTIONSqrt{\Zpower{(x-a)}{3}}
41 \Zpower{(x-b)}{3}
42 \Zpower{(x-c)}{3}}}}
43 %
44 \ZDefrhsONE{
45 =\Zfrac{2}{\Zpower{(a-b)}{2}}
46 \Zpower{(b-c)}{2} \ZFUNCTIONSqrt{\Zpower{(a-c)}{3}}}}
47 %
48 \ZDefrhsTWO{
49 \ZContinuationSpacing
50 \times
51 \Zsurbrack{(a-b)(2a-b-c)}
52 \ZFUNCTIONArg{\ZFUNCTIONEllipticFunctionFz}{(\nu, q)}-2
53 \Zsurround{\Zpower{a}{2}+\Zpower{b}{2}+\Zpower{c}{2}-ab-ac-bc}
54 \ZFUNCTIONArg{\ZFUNCTIONEllipticFunctionEz}{(\nu, q)}}
55 %
56 \ZDefrhsTHREE{
57 \ZContinuationSpacing
58 +\Zfrac{2[u(a+b-2c)-a(a-c)-b(b-c)]}{\Zpower{(a-b)}{2}
59 (a-c)(b-c)\ZFUNCTIONSqrt{(u-a)(u-b)(u-c)}}
60 }
61 %
62 \ZDefConstraint{ \Zsurbrack{u>a>b>c}}
63 %
64 \ZDefReference{BY (238.16)}
65 %
66 \ZZdoit
67
68 \ZendE

```

5.2 Formatted output

Is formatted as follows:

3.136

$$\begin{aligned}
1. \quad & \int_{-\infty}^u \frac{dx}{\sqrt{(a-x)^3(b-x)^3(c-x)^3}} \\
&= \frac{2}{(a-b)^2(b-c)^2\sqrt{(a-c)^3}} \\
&\quad \times [(b-c)(a+b-2c)F(\alpha, p) - 2(c^2 + a^2 + b^2 - ab - ac - bc)E(\alpha, p)] \\
&\quad + \frac{2[c(a-c) + b(a-b) - u(2a-c-b)]}{(a-b)(a-c)(b-c)^2\sqrt{(a-u)(b-u)(c-u)}} \\
&\hspace{15em} [a > b > c > u] \hspace{10em} \text{BY (231.14)}
\end{aligned}$$

$$\begin{aligned}
2. \quad & \int_u^{\infty} \frac{dx}{\sqrt{(x-a)^3(x-b)^3(x-c)^3}} \\
&= \frac{2}{(a-b)^2(b-c)^2\sqrt{(a-c)^3}} \\
&\quad \times [(a-b)(2a-b-c)F(\nu, q) - 2(a^2 + b^2 + c^2 - ab - ac - bc)E(\nu, q)] \\
&\quad + \frac{2[u(a+b-2c) - a(a-c) - b(b-c)]}{(a-b)^2(a-c)(b-c)\sqrt{(u-a)(u-b)(u-c)}} \\
&\hspace{15em} [u > a > b > c] \hspace{10em} \text{BY (238.16)}
\end{aligned}$$

6 System level details of GR6

6.1 L^AT_EX packages

The L^AT_EX packages that were used are:

- `graphicx` (needed for its `\rotatebox` command)
- `amsmath` (for the fonts)
- `amsmath` (for mathematics)
- `comment` (to comment out sections)
- `longtable` (for run-on tables)
- `multicol` (for multi-column material)
- `supertabular` (for run-on tables)
- `type1cm` (state intention to use type 1 fonts)
- `url` (for printing URL's properly)

6.2 Directories and files

GR6 has two directories of information:

- `GR/FINAL`: where all the final book content appears (source for GR6)
- `GR/STYLES`: the location of the L^AT_EX style files

The search path for T_EX included the directory in `GR/STYLES`.

6.2.1 GR/STYLES files

The files in `GR/STYLES` are all standard T_EX/L^AT_EX files, and include:

- Ueuf57.fd
- Umsa.fd
- Umsa57.fd
- Umsb.fd
- Umsb57.fd
- amsbsy.sty
- amsfonts.sty
- amsgen.sty
- amsmath.sty
- amsopn.sty
- amssymb.sty
- amstext.sty
- bk10.clo
- book.cls
- cmdtrack.sty
- comment.sty
- fullpage.sty
- index.sty
- lastpage.sty
- longtable.sty
- minitoc.sty.gz
- multicol.sty
- oldgerm.sty
- supertabular.sty
- times.sty
- type1cm.sty
- ueuf57.fd
- umsa.fd
- umsa57.fd
- umsb.fd
- umsb57.fd
- url.sty
- uyfrak.fd

6.2.2 GR/FINAL files

6.2.2.1 GR6 source There is one source file for each section and chapter of GR6 (when arranged in alphabetical order they create GR6 in the proper order). The source files are:

- | | |
|---|---|
| <ol style="list-style-type: none"> 1. chapter.0.0.0.title_page.tex.tex 2. chapter.0.0.1.tab_contents.tex.tex 3. chapter.0.0.2.preface.tex.tex 4. chapter.0.0.3.acknowledge.tex.tex 5. chapter.0.3.order_present.tex.tex 6. chapter.0.4.using_tables.tex.tex 7. chapter.0.5.special_funct.tex.tex 8. chapter.0.6.notations.tex.tex 9. chapter.0.7.note_on_bib_ref.tex.tex 10. chapter.00.all.introduction.tex.tex 11. chapter.01.all.tex.tex 12. chapter.02.indef_elem_funcs.tex.tex 13. chapter.03.def_elem_funcs_1.tex.tex 14. chapter.04.def_elem_funcs_2.tex.tex 15. chapter.05.indef_specfn.tex.tex 16. chapter.06.def_spec_funcs_1.tex.tex | <ol style="list-style-type: none"> 17. chapter.07.def_spec_funcs_2.tex.tex 18. chapter.08.spec_funcs_1.tex.tex 19. chapter.09.spec_funcs_2.tex.tex 20. chapter.10.vec_field_theory.tex.tex 21. chapter.11.alg_inequalities.tex.tex 22. chapter.12.int_inequalities.tex.tex 23. chapter.13.matrices.tex.tex 24. chapter.14.determinants.tex.tex 25. chapter.15.norms.tex.tex 26. chapter.16.diff_eqn.tex.tex 27. chapter.17.int_transforms.tex.tex 28. chapter.18.z_transform.tex.tex 29. chapter.19.references.tex.tex 30. chapter.20.suppl_references.tex.tex 31. chapter.Z.index.tex.tex |
|---|---|

6.2.2.2 The shell scripts The shell scripts include scripts for manipulating the index (putting in *continued* at the top of continuation pages, and making page breaks as appropriate)::

- sh.modify_concept_index.pl
- sh.modify_function_index.pl

and the single script that runs everything (including L^AT_EX and the above two shell scripts), `doit.gr.all`:

```
#-----
# run LaTeX on the entire book
#-----
latex gr.tex.all.tex

#-----
# Make the indices
# Create the files
#       .nfx --> .nfd
#       .ncx --> .ncd
#-----
perl -pi.bak -e 's/ZdddZ/\$/g;' gr.tex.all.nfx
perl -pi.bak -e 's/ZsssZ/\$/g;' gr.tex.all.nfx
perl -pi.bak -e 's/ZbbbZ/\$/g;' gr.tex.all.nfx
perl -pi.bak -e 's/ZpppZ/\^/g;' gr.tex.all.nfx
```

```
perl -pi.bak -e 's/Schlafli/Schl{\f\\a}fli/g;' gr.tex.all.nfx
perl -pi.bak -e 's/Schlafli/Schl{\f\\a}fli/g;' gr.tex.all.nfd
perl -pi.bak -e 's/Schlafli/Schl{\f\\a}fli/g;' gr.tex.all.nfc

makeindex -s gr.ist -o gr.tex.all.ncd gr.tex.all.ncx # concept index
makeindex -s gr.ist -o gr.tex.all.nfd gr.tex.all.nfx # functions

#-----
# fix the font on the indices (to make it be \sffamily)
#-----
perl -pi.bak -e 's/(Function and constant index)/\sffamily $1/;' gr.tex.all.toc
perl -pi.bak -e 's/(General index)/\sffamily $1/;' gr.tex.all.toc
```

6.2.2.3 Other files

The file `gr.ist` has the format for the index

```
% This is the index style file
% It is modeled after the LaTeX Companion (page 361)
%-----

% this puts a large letter A, B, etc before the appropriate section
%-----
heading_prefix "{\Large\bfseries\hfill "
heading_suffix "\hfill}\nopagebreak\n "
headings_flag 1

% this puts leading dots to the page numbers
%-----
delim_0 "\dotfill "
delim_1 "\dotfill "
delim_2 "\dotfill "
```

There are 11 figures in the book, each is an `.eps` file. Each figure file is numbered by where it appears in the book. The figures are:

- fig.4.611.1.eps
- fig.4.611.2.eps
- fig.4.612.1.eps
- fig.4.612.2.eps
- fig.4.613.1.eps
- fig.4.613.2.eps
- fig.4.613.3.eps
- fig.4.613.4.eps
- fig.4.614.eps
- fig.4.615.eps
- fig.4.616.eps
- fig.8.310.2.eps
- fig.8.412.6.eps
- fig.8.422.2.eps
- fig.8.423.2.eps
- fig.9.512.eps