# Subject Index

## A
- **ABORT statement**
  - exiting IML, 474
- **ABS statement**
  - absolute value, 474
- **ALL statement**
  - checking for nonzero elements, 474
- **ANY function**
  - checking for nonzero elements, 475
- **APPEND statement**
  - SAS data sets, 476
- **APPLY function**, 478
- **ARMACOV call**
  - autocovariance sequence, 479
- **ARMALIK call**
  - log likelihood and residuals, 482
- **ARMASIM function**
  - simulating univariate ARMA series, 484
- **Bessel function**
  - finding nonzero roots and derivatives of, 577–578
- **BLOCK function**
  - forming block-diagonal matrices, 486
- **box-and-whisker plots**, 822
- **BRANKS function**
  - computing bivariate ranks, 487
- **BTRAN function**
  - computing the block transpose, 488
- **BYTE function**
  - translating numbers to ordinal characters, 489
- **CALL statement**
  - calling a subroutine or function, 490
- **CHANGE call**
  - replacing text in an array, 490
- **CHAR function**
  - character representation of a numeric matrix, 491
- **Character Functionality**, 450
- **CHOOSE function**
  - choosing and changing elements, 492
- **CLOSE statement**
  - closing SAS data sets, 493
- **CLOSEFILE statement**
  - closing a file, 494
- **CONCAT function**
  - performing elementwise string concatenation, 496
- **CONTENTS function**
  - obtaining the variables in SAS data sets, 497
- **Control Statements**, 454
- **CONVEXIT function**
  - calculating convexity of non contingent “cash flows”, 498
- **CONVMOD function**
  - converting modules to character matrices, 498
- **correlation coefficient computation**, 822
- **COVLAG function**
  - computing autocovariance estimates, 499
- **CREATE statement**
  - creating new SAS data sets, 500
- **CSHAPE function**
  - reshaping and repeating character values, 502
- **CUSUM function**
  - calculating cumulative sums, 504
- **CVEXHULL function**
  - finding a convex hull, 505

## B
- **B**
  - Bessel function
    - finding nonzero roots and derivatives of, 577–578
  - **BLOCK function**
    - forming block-diagonal matrices, 486
  - **box-and-whisker plots**, 822
  - **BRANKS function**
    - computing bivariate ranks, 487
  - **BTRAN function**
    - computing the block transpose, 488
  - **BYTE function**
    - translating numbers to ordinal characters, 489

## C
- **CALL statement**
  - calling a subroutine or function, 490
- **CHANGE call**
  - replacing text in an array, 490
- **CHAR function**
  - character representation of a numeric matrix, 491
- **Character Functionality**, 450
- **CHOOSE function**
  - choosing and changing elements, 492
- **CLOSE statement**
  - closing SAS data sets, 493
- **CLOSEFILE statement**
  - closing a file, 494
- **CONCAT function**
  - performing elementwise string concatenation, 496
- **CONTENTS function**
  - obtaining the variables in SAS data sets, 497
- **Control Statements**, 454
- **CONVEXIT function**
  - calculating convexity of non contingent “cash flows”, 498
- **CONVMOD function**
  - converting modules to character matrices, 498
- **correlation coefficient computation**, 822
- **COVLAG function**
  - computing autocovariance estimates, 499
- **CREATE statement**
  - creating new SAS data sets, 500
- **CSHAPE function**
  - reshaping and repeating character values, 502
- **CUSUM function**
  - calculating cumulative sums, 504
- **CVEXHULL function**
  - finding a convex hull, 505

## D
- **Dataset and File Functionality**, 454
- **DATASETS function**
  - obtaining names of SAS data sets, 505
- **DELETE call**
  - deleting SAS data sets, 508
- **DELETE statement**
  - marking observations for deletion, 506
- **DESIGN function**
  - creating a design matrix, 509
- **DESIGNF function**
  - creating a full-rank design matrix, 510
- **DET function**
  - computing determinants of a square matrix, 510
- **DIAG function**
  - creating a diagonal matrix, 511
- **DISPLAY statement**
  - displaying fields in display windows, 512
- **DO and END statements**
  - grouping statements as a unit, 514
- **DO DATA statement**
  - repeating a loop until, 516
- **DO function**
  - producing an arithmetic series, 513
- **DO statement**
  - DATA clause, 516
  - UNTIL clause, 517
  - WHILE clause, 517
- **DO statement, iterative**
  - iteratively executing a DO group, 515
DO UNTIL statement
conditionally executing statements iteratively, 517
DO WHILE statement
conditionally executing statements iteratively, 517
DURATION function
calculating modified duration of non-contingent cashflows, 518

E
ECHELON function
reducing a matrix to row-echelon normal form, 519
EDIT statement
opening a SAS data set for editing, 520
EIGEN call
computing eigenvalues and eigenvectors, 522
eigenvalue decomposition
compared with ODE call, 670
EIGVAL function
computing eigenvalues, 525
EIGVEC function
computing right eigenvectors, 526
END statement
ending a DO loop or DO statement, 526
EXECUTE call
executing SAS statements immediately, 527
EXP function
calculating the exponential, 527

F
FFT function
performing the finite Fourier transform, 527
FILE statement
opening or pointing to an external file, 528
FIND statement
finding observations, 529
FINISH statement
denoting the end of a module, 531
forward rates, 532
FREE statement
freeing matrix storage space, 532

G
GBLKVP call
defining a blanking viewport, 533
GBLKVPD call
deleting the blanking viewport, 534
GCLOSE call
closing the graphics segment, 535
GDELETE call
deleting a graphics segment, 535
GDRAW call
drawing a polyline, 535
GDRAWL call
drawing individual lines, 536
GENEIG call
generalized eigenproblems, 537
GGRID call
drawing a grid, 538
GINCLUDE call
including graphics segments, 539
GINV function
computing generalized inverses, 540
GOPEN call
opening graphics segments, 541
GOTO statement
jumping to a new statement, 541
GPIE call
drawing pie slices, 542
GPIEXY call
converting coordinates, 543
GPOINT call
ploting points, 545
GPOLY call
drawing and filling a polygon, 546
GPORT call
defining a viewport, 547
GSPORTPOP call
popping viewports, 548
GSPORTSTK call
stacking viewports, 548
Graphics and Window Functions, 455
GSORT call
calculating round numbers for labeling axes, 548
GSCRIPT call
writing multiple text strings, 549
GSET call
setting attributes for graphics segments, 550
GSHOW call
showing a graph, 551
GSORT call
computing the Gram-Schmidt orthonormalization, 552
GSTART call
initializing the graphics system, 553
GSTOP call
deactivating the graphics system, 554
GSTRLEN call
finding the string length, 554
GTTEXT and GVTEXT calls
placing text on a graph, 555
GWIND ge call
defining the data window, 556
GXAXIS and GYAXIS calls
drawing an axis, 557

H
HALF function
computing Cholesky decomposition, 559
HANKEL function
generating a Hankel matrix, 559
HDIR function
performing a horizontal direct product, 561
HERMITE function
reducing a matrix to Hermite normal form, 562
HOMOGEN function
solving homogeneous linear systems, 563
### Subject Index

<table>
<thead>
<tr>
<th>Function/Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I function</td>
<td>creating an identity matrix, 564</td>
</tr>
<tr>
<td>IF-THEN/ELSE</td>
<td>conditionally executing statements, 564</td>
</tr>
<tr>
<td>FFT function</td>
<td>computing the inverse finite Fourier transform, 566</td>
</tr>
<tr>
<td>IMLMLIB Module Library</td>
<td>modules reference, 822–824, 826–827, overview, 819–821</td>
</tr>
<tr>
<td>INDEX statement</td>
<td>indexing a variable in a SAS data set, 567</td>
</tr>
<tr>
<td>INFILE statement</td>
<td>opening a file for input, 567</td>
</tr>
<tr>
<td>INPUT statement</td>
<td>inputing data, 569</td>
</tr>
<tr>
<td>INSERT function</td>
<td>inserting one matrix inside another, 570</td>
</tr>
<tr>
<td>INT function</td>
<td>truncating a value, 571</td>
</tr>
<tr>
<td>INV function</td>
<td>computing a matrix inverse, 572</td>
</tr>
<tr>
<td>J function</td>
<td>creating a matrix of identical values, 576</td>
</tr>
<tr>
<td>K function</td>
<td>Kalman filter subroutines examples, 279</td>
</tr>
<tr>
<td>LABEL</td>
<td>quadratic form maximization, 620</td>
</tr>
<tr>
<td>LCP call</td>
<td>solving the linear complementarity problem, 597</td>
</tr>
<tr>
<td>least absolute value regression, 592, 594–596</td>
<td></td>
</tr>
<tr>
<td>LENGTH function</td>
<td>finding the lengths of character matrix elements, 598</td>
</tr>
<tr>
<td>Linear Algebra Functionality, 452</td>
<td></td>
</tr>
<tr>
<td>linear least squares</td>
<td>full rank example, 696–697</td>
</tr>
<tr>
<td>QR decomposition, 711</td>
<td></td>
</tr>
<tr>
<td>rank deficient solutions, 728, 731–732</td>
<td></td>
</tr>
<tr>
<td>LIST statement</td>
<td>jumping to another statement, 599</td>
</tr>
<tr>
<td>LMS and LTS calls</td>
<td>performing robust regression, 602</td>
</tr>
<tr>
<td>LOAD statement</td>
<td>loading modules and matrices, 609</td>
</tr>
<tr>
<td>LOC function</td>
<td>finding nonzero elements of a matrix, 610</td>
</tr>
<tr>
<td>LOG function</td>
<td>taking the natural logarithm, 611</td>
</tr>
<tr>
<td>LP call</td>
<td>solving the linear programming problem, 612</td>
</tr>
<tr>
<td>LTS call</td>
<td>performs robust regression, 614</td>
</tr>
<tr>
<td>LUPDT call</td>
<td>615</td>
</tr>
<tr>
<td>MAD function</td>
<td>univariate median-absolute-deviation, 615</td>
</tr>
<tr>
<td>MARG call</td>
<td>evaluating marginal totals, 616</td>
</tr>
<tr>
<td>matrix decomposition</td>
<td>Cholesky decomposition, 728, 730</td>
</tr>
<tr>
<td>complete orthogonal decomposition, 475–476, 494–496</td>
<td></td>
</tr>
<tr>
<td>downdating and updating, 709–713, 728</td>
<td></td>
</tr>
<tr>
<td>QR decomposition, 676, 678–680, 692–697</td>
<td></td>
</tr>
<tr>
<td>Matrix Inquiry Functions, 449</td>
<td></td>
</tr>
<tr>
<td>Matrix Reshaping Functions, 450</td>
<td></td>
</tr>
<tr>
<td>MATTRIB statement</td>
<td>associating printing attributes with matrices, 617</td>
</tr>
<tr>
<td>MAX function</td>
<td>finding the maximum value of matrix, 619</td>
</tr>
<tr>
<td>MCD and MVE calls</td>
<td>621</td>
</tr>
<tr>
<td>median computation, 824</td>
<td></td>
</tr>
<tr>
<td>MIN function</td>
<td>finding the smallest element of a matrix, 626</td>
</tr>
<tr>
<td>MOD function</td>
<td>computing the modulo (remainder), 627</td>
</tr>
<tr>
<td>NAME function</td>
<td>listing the names of arguments, 627</td>
</tr>
<tr>
<td>NCOL function</td>
<td>finding the number of columns of a matrix, 628</td>
</tr>
<tr>
<td>NLENG function</td>
<td>finding the size of an element, 628</td>
</tr>
<tr>
<td>nonlinear optimization subroutines</td>
<td>advanced examples, 335</td>
</tr>
<tr>
<td>conjugate gradient optimization, 632</td>
<td></td>
</tr>
<tr>
<td>control parameters vector, 331–333</td>
<td></td>
</tr>
<tr>
<td>double dogleg optimization, 633, 636</td>
<td></td>
</tr>
<tr>
<td>feasible point computation, 639</td>
<td></td>
</tr>
<tr>
<td>finite difference approximations, 313–315, 636–637, 639</td>
<td></td>
</tr>
<tr>
<td>global vs. local optima, 305</td>
<td></td>
</tr>
<tr>
<td>hybrid quasi-Newton optimization, 640, 642, 644</td>
<td></td>
</tr>
<tr>
<td>Kuhn-Tucker conditions, 306</td>
<td></td>
</tr>
</tbody>
</table>
least-squares methods, 640, 642, 644, 646  
Levenberg-Marquardt optimization, 644, 646  
Nelder-Mead simplex optimization, 646–648, 650  
Newton-Raphson optimization, 650, 652  
Newton-Raphson ridge optimization, 652, 654  
objective function and derivatives, 308–313  
options vector, 318–320, 322–324  
parameter constraints, 315, 317–318  
printing optimization history, 333, 335  
quadratic optimization, 659, 661, 663  
quasi-Newton optimization, 654–655, 657, 659  
return codes, 307  
termination criteria, 324–329, 331  
trust-region optimization, 663–664  
NORMAL function  
generating a pseudo-random normal deviate, 664  
NROW function  
finding the number of rows of a matrix, 664  
NUM function  
producing a numeric representation of a character  
matrix, 665  
Numerical Analysis Functionality, 452  
umerical integration, 697–703  
adaptive Romberg method, 699  
of differential equations, 665, 667, 669–670, 672–  
673  
specifying subintervals, 698  
two-dimensional integration, 702  
O  
OPSCAL Function, 673  
Optimization Subroutines, 453  
ORPOL function  
generating orthogonal polynomials, 675  
orthogonal factorization, 728, 730  
orthogonalization  
by ORTVEC call, 676, 678–680  
P  
PARSE statement  
parsing matrix elements as statements, 680  
PAUSE statement  
interrupting module execution, 681  
PGRAF call  
producing scatter plots, 682  
POLYROOT function  
finding zeros of a real polynomial, 683  
PRINT statement  
printing matrix values, 684  
probability contour plot, 823  
PRODUCT function  
multiplying matrices of polynomials, 686  
PURGE statement  
removing observations marked for deletion, 687  
PUSH call, 688  
PUT statement  
writing data to an external file, 689  
PV function  
calculating present value, 691  
Q  
quadratic form maximization, 620–621  
 quartile computation, 825  
QUEUE call  
queueing SAS statements, 704  
QUIT statement  
exiting from IML, 705  
R  
RANK function  
ranking elements of a matrix, 705  
RANKTIE function  
ranking matrix elements using tie-averaging, 706  
RATES function  
converting interest rates, 707  
RATIO function  
dividing matrix polynomials, 707  
READ statement  
reading observations from a data set, 713  
Reduction Functions, 449  
regression, 825  
best subsets, 621  
least absolute value, 592, 594–596  
response surface, 824  
REMOVE function  
discarding elements from a matrix, 717  
REMOVE statement  
removing matrices from storage, 718  
RENAME call  
renaming SAS data sets, 718  
REPLACE statement  
replacing values, 719  
RESET statement  
setting processing options, 722  
reshaping matrices, 822, 826  
response surface regression, 824  
RESUME statement  
resuming execution, 724  
RETURN statement  
returning to caller, 724  
ROOT function  
performing the Cholesky decomposition of a ma- 
trix, 725  
ROWCAT function  
concatenating rows without blank compression,  
725  
ROWCATC function  
concatenating rows with blank compression, 726  
RUN statement  
executing statements in a module, 727  
S  
SAVE statement  
saving data, 733  
Scalar Functions, 449
scatter plots, 823
sequential tests, 734–740, 742–744, 746–748
group sequential methods, 742–743
minimizing average sample number (ASN), 746–748
randomized clinical trials, 746–748
scaling, 736
shifting, 737
Set Functions, 453
SETDF function
comparing elements of two matrices, 749
SETIN statement
making a data set current for input, 749
SETOUT statement
making a data set current for output, 750
SHAPE function
reshaping and repeating values, 751
SHOW statement
printing system information, 752
SOLVE function
solving a system of linear equations, 753
SORT statement
sorting a SAS data set, 754
SOUND call
producing a tone, 755
SPLINE call
evaluating points on the spline, 756
SPLINEC call
evaluating points on the spline, 758
splines, 760, 762–763
integration of splines, 762
SPLINEV function
evaluating points on a spline, 759
SPOT function
calculating spot rates, 764
SQRSYM function
converting to a square matrix, 764
SQRT function
calculating the square root, 765
SSQ function
calculating the sum of squares, 765
standardizing numeric data, 827
START and FINISH statements
defining a module, 766
Statistical Functionality, 450
STOP statement
stopping execution of statements, 766
STORE function
listing names of matrices and modules, 767
STORE statement
storing matrices and modules, 767
SUBSTR function
taking substrings of matrix elements, 768
substring replacement, 826
SUM function
summing all elements, 769
SUMMARY statement
computing summary statistics, 769
SVD call
computing the singular value decomposition, 773
SWEEP function
sweeping a matrix, 774
SYMSQR function
converting to a symmetric matrix, 776
T
T function
transposing a matrix, 776
TEIGEN call
computing eigenvalues and eigenvectors, 777
TEIGVAL functions
computing eigenvalues, 777
TEIGVEC functions
computing eigenvectors, 777
time series analysis and control
AR model selection, 225–227, 804
ARMA model prediction, 245–246, 801
Bayesian constrained least squares, 265–267
Bayesian seasonal adjustment, 242–244, 254–255, 789, 791
instantaneous response model, 229–230, 274–276
ISM TIMSAC packages, 270–272
least squares and Householder transformation, 264–265
locally stationary multivariate time series, 796, 798
locally stationary time series, 795–796
minimum AIC method, 225–230, 250–252
missing values, 269
multivariate time series, 245–246, 259–261, 801
nonstationary covariance function analysis, 803
nonstationary data analysis, 231, 233, 235–241
nonstationary time series, 256–257, 259, 791, 793, 795
overview, 223
periodic AR model, 800–801
roots of AR and MA equations, 247–248, 802
smoothness priors modeling, 252–253, 791, 793, 795
spectral analysis, 261–264
state space and Kalman filter method, 267–269
VAR model, 228–230, 273–275, 798, 800
Time Series Functionality, 451
TOEPLITZ function
generating a Toeplitz matrix, 777
TPSPLINE call
computing thin-plate smoothing splines, 779
TPSPLNEV call
evaluating thin-plate smoothing splines, 783
TRACE function
summing diagonal elements, 788
triangular linear systems, 788
TYPE function
determining matrix types, 806
U
UNIFORM function
generating pseudo-random uniform deviates, 807
UNION function
   performing unions of sets, 807
UNIQUE function
   sorting and removing duplicates, 808
USE statement
   opening SAS data sets, 808

V
VALSET call
   performing indirect assignments, 810
VALUE function
   assigning values, 811
VECDIAG function
   creating vectors, 811

W
WINDOW statement
   opening a display window, 812

X
XMULT function
   performing accurate matrix multiplication, 814
XSECT function
   intersecting sets, 815

Y
YIELD function
   calculating yield-to-maturity of a cash flow streams, 815
Syntax Index

A
ABORT statement, 474
ABS statement, 474
ALL statement, 474
ANY function, 475
APPCORT call, 475–476
APPEND statement, 476
APPLY function, 478
ARMACOV call, 479
ARMALIK call, 482
ARMASIM function, 484

B
BLOCK function, 486
BRANKS function, 487
BTRAN function, 488
BYTE function, 489

C
CALL statement, 490
CHANGE call, 490
CHAR function, 491
CHOOSE function, 492
CLOSE statement, 493
CLOSEFILE statement, 494
COLVEC function, 822
COMPOR T call, 494–496
CONCAT function, 496
CONTENTS function, 497
CONVEXIT function, 498
CONVMOD function, 498
CORR function, 822
COVLAG function, 499
CREATE statement, 500
CSHAPE function, 502
CUSUM function, 504
CVEXHULL function, 505

D
DATASETS function, 505
DELETE call, 508
DELETE statement, 506
DESIGN function, 509
DESGNF function, 510
DET function, 510
DIAG function, 511
DISPLAY statement, 512
DO and END statements, 514
DO DATA statement, 516
DO function, 513
DO statement, iterative, 515
DO UNTIL statement, 517
DO WHILE statement, 517
DURATION function, 518

E
ECHELON function, 519
EDIT statement, 520
EIGEN call, 522
EIGVAL function, 525
EIGVEC function, 526
END statement, 526
EXECUTE call, 527
EXP function, 527

F
FFT function, 527
FILE statement, 528
FIND statement, 529
FINISH statement, 531
FORWARD function, 532
FREE statement, 532

G
GBLKVP call, 533
GBLKVD call, 534
GBXWHSKR call, 822
GCLOSE call, 535
GDELETE call, 535
GDRAW call, 535
GDRAWL call, 536
GENEIG call, 537
GRID call, 538
GINCLUDE call, 539
GINV function, 540
GOPEN call, 541
GOTO statement, 541
GPIE call, 542
GPIEXY call, 543
GPOINTER call, 545
GPOLY call, 546
GPORT call, 547
GPORTPOP call, 548
GPORTSTK call, 548
GPROBCNT call, 823
GSCLALE call, 548
GSCRIPT call, 549
GSET call, 550
GSHOW call, 551
GSORTH call, 552
GSTART call, 553
GSTOP call, 554
GSTRLEN call, 554
GTEXT and GVTEXT calls, 555
GWINDOW call, 556
GXAXIS and GYAXIS calls, 557
GXYPLOT call, 823

H
HALF function, 559
HANKEL function, 559
HDIR function, 561
HERMITE function, 562
HOMOGEN function, 563

I
I function, 564
IF-THEN/ELSE statement, 564
IFFT function, 566
IMLMLIB Module Library
modules reference, 822–824, 826–827
overview, 819–821
INDEX statement, 567
INFILE statement, 567
INPUT statement, 569
INSERT function, 570
INT function, 571
INV function, 572
INVUPDT function, 573
IPF call, 574

J
J function, 576
JROOT function, 577–578

K
KALCVS call, 285, 581–584
KALDF call, 288, 584, 586–588
KALDFS call, 589–590
Kalman filter subroutines
examples, 279
overview, 278–279
syntax, 578

L
LAV call, 592, 594–596
LCP call, 597
LENGTH function, 598
LINK and RETURN statements, 599
LIST statement, 600
LMS and LTS calls, 602
LOAD statement, 609
LOC function, 610
LOG function, 611
LP call, 612
LTS call, 614
LUPDT call, 615

M
MAD function, 615
MARG call, 616
MATTRIB statement, 617
MAX function, 619
MAXQFORM call, 620–621
MCD and MVE calls, 621
MEDIAN function, 824
MIN function, 626
MOD function, 627

N
NAME function, 627
NCOL function, 628
NLENG function, 628
nonlinear optimization subroutines
advanced examples, 335
details, 305
introductory examples, 297
NLPCG Call, 340
NLPCG call, 632
NLPDD Call, 345, 347, 364, 633
NLPDD call, 633
NLPDDCall, 636
NLPFDDCall, 353, 636
NLPFDD call, 636–637
NLPFDD Call, 639
NLPFEACall, 639
NLPHQNCall, 640
NLPHQN Call, 642, 644
NLPMLCall, 355, 644
NLPML Call, 644
NLPML Call, 646
NLPNMS call, 646–647
NLPNMS Call, 648, 650
NLPNRA Call, 650
NLPNRA call, 650
NLPNRA Call, 652
NLPNRACall, 341
NLPNRACall, 652
NLPNRCall, 654
NLPQNCall, 349–350, 368
NLPQNCall, 654
NLPQNCall, 655, 657, 659
NLPQUA call, 659
NLPQUACall, 661, 663
NLPRTR Call, 337, 352
NLPRTR call, 663
NLPRTR Call, 664
overview, 295
syntax, 628
NORMAL function, 664
NROW function, 664
NUM function, 665

O
ODE call, 665, 667, 669–670, 672–673
OPSCAL function, 673
ORPOL function, 675
ORTVEC call, 676, 678–680

P
PARSE statement, 680
PAUSE statement, 681
PGRAF call, 682
POLYROOT function, 683
PRINT statement, 684
PRODUCT function, 686
PURGE statement, 687
PUSH call, 688
PUT statement, 689
PV function, 691

Q
QR call, 692–697
QUAD call, 697–703
QUADREG call, 824
QUARTILE function, 825
QUEUE call, 704
QUIT statement, 705

R
RANK function, 705
RANKTIE function, 706
RATES function, 707
RATIO function, 707
RDODT call, 709–713
READ statement, 713
REGRESS call, 825
REMOVE function, 717
REMOVE statement, 718
RENAMF call, 718
REPEAT function, 719
REPLACE statement, 719
RESET statement, 722
RESUME statement, 724
RETURN statement, 724
ROOT function, 725
ROWCAT function, 725
ROWCATC function, 726
ROWVEC function, 826
RSUBSTR function, 826
RUN statement, 727
RUPDT call, 709–713, 728
RZLIND call, 728, 730–733

S
SAVE statement, 733
SEQ call, 734–740, 742–744, 746–748
SEQSCALE call, 734–740, 742–744, 746–748
SEQSHIFT call, 734–740, 742–744, 746–748
SETDIFF function, 749
SETIN statement, 749
SETOUT statement, 750
SHAPE function, 751
SHOW statement, 752
SOLVE function, 753
SORT statement, 754
SOUND call, 755
SPLINE call, 756, 760, 762–763
SPLINEC call, 758, 760, 762–763
SPLINEV function, 759, 762–763
SPOT function, 764
SQR function, 764
SQRMS function, 764
SQR function, 765
SSQ function, 765
START and FINISH statements, 766
STOP statement, 766
STORAGE function, 767
STORE statement, 767
SUBSTR function, 768
SUM function, 769
SUMMARY statement, 769
SVD call, 773
SWEEP function, 774
SYMSQR function, 776

T
T function, 776
TABPRT call, 827
TEIGEN call, 777
TEIGVAL functions, 777
TEIGVEC functions, 777
TIMSAC subroutines
advanced examples, 273
details, 250
introductory examples, 225
overview, 223
syntax, 249
TSBAYSEA subroutine, 242–244, 789, 791
TSDECOMP subroutine, 241, 791, 793, 795
TSMLOCAR subroutine, 231, 795–796
TSMLOCAR subroutine, 239, 796, 798
TSMULMAR subroutine, 228–230, 273–276, 798,
800
TSPEARS subroutine, 800–801
TSPRED subroutine, 245–246, 275, 801
TSROOT subroutine, 247–248, 802
TSTVCAR subroutine, 803
TSUNIMAR subroutine, 804
TOEPLITZ function, 777
TPSPLINE call, 779
TPSPLNEV call, 783
TRACE function, 788
TRISOLV function, 788
TYPE function, 806

U
UNIFORM function, 807
UNION function, 807
UNIQUE function, 808
USE statement, 808

V
VALSET call, 810
VALUE function, 811
VECDIAG function, 811

W
WINDOW statement, 812

X
XMULT function, 814
XSECT function, 815

Y
YIELD function, 815